AFIT/GA/ENY/91D-9

APPLICATION OF LOW THRUST PROPULSION TECHNIQUES TO SATELLITE ATTITUDE CONTROL SYSTEMS

THESIS

Jeffrey J. Abbott, Captain, USAF
AFIT/GA/ENY/91D-9

AFIT/GA/ENY/91D-9

APPLICATION OF LOW THRUST PROPULSION TECHNIQUES TO SATELLITE ATTITUDE CONTROL SYSTEMS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Astronautical Engineering

Jeffrey J. Abbott, B.S.

Captain, USAF

November, 1991

Preface

The purpose of this study is to investigate the effectiveness of low-thrust propulsion techniques in satellite attitude control. Of particular interest are any differences in attitude control system design methods that make use of the advantages of low-thrust propulsion, such as small minimum impulse bits, high frequency of operation, or controllability.

I would like to thank Capt. Jim Planeaux, who assisted me and advised me during the initial stages of this project. I would also like to thank Dr. Wiesel, who then assumed his duties as my advisor, and gave me the guidance and questioning I needed to accomplish my goal. Finally, I am grateful to my classmates, who found the time to share their knowledge and expertise.

Table of Contents

List of Figures

List of Symbols

A	A Frame, nominal attitude frame; also area
A_x, A_y, A_z	Axis of A Frame
A_{eff}	Effective Area, projection of area onto solar vector
c	Speed of light
db	Deadband, error pointing limits
dM	Differential element of mass
e	Orbit eccentricity
F	Force vector
$\mathbf{F}_a$	Force vector due to absorbed solar radiation
$\mathbf{F}_g$	Force vector due to gravity
$\mathbf{F}_r$	Force vector due to reflected solar radiation
$\mathbf{F}_s$	Net force vector due to solar radiation
$\mathbf{F}_t$	Force vector due to radio frequency transmissions
H	Spacecraft angular momentum vector
$\mathbf{H}_b$	Body of spacecraft angular momentum vector
h	Angular momentum of a momentum wheel
h	Angular momentum contained in momentum wheel system
I	Moment of Inertia
[I]	Moments of Inertia matrix
i	Orbit inclination
i, j, k	Unit vectors in body frame
K_a	Feedback system gain for element about 'a' axis
M	Net moment vector
$\mathbf{M}_c$	Control moment vector
$\mathbf{M}_g$	Gravitational moment vector
$\mathbf{M}_s$	Solar pressure moment vector
$\mathbf{M}_t$	Radio frequency pressure moment vector
N	Vector normal to orbital plane
N_m	Nutation frequency damping factor
n	Vector normal to surface of interest
P	Solar radiation pressure at 1 AU
P_i	Power input to propulsion system
P_t	Power transmitted
[R]	Rotation matrix of body frame with respect to A frame
R	Position vector of point with respect to Earth center
$\mathbf{R}_e$	Position vector of point on Earth's surface
$\mathbf{R}_o$	Position vector of spacecraft center of mass
r	Position vector of point with respect to spacecraft center
S	Solar radiation direction vector
V_e	Exhaust velocity
α	Orbit angle as measured from spacecraft noon
γ	Angle between subpoint vector and spacecraft position vector

Δt	Step size
δ	Solar declination
ε	Vector of Pointing errors in roll, pitch and yaw
ζ	Thruster offset angle for control method 2
η	Propulsion system efficiency
θ	Pointing error about pitch axis
λ	Angle between desired **y** vector and spacecraft position vector
μ_e	Gravitational constant times mass of Earth
ν	True anomaly
o	'Higher order terms'
ρ	Reflectivity
τ	Time constant for controller feedback
ϕ	Pointing error about roll axis
ψ	Pointing error about yaw axis
ω, $\omega^{B/I}$	Angular velocity vector of body frame with respect to inertial space
$\omega^{B/A}$	Angular velocity vector of body frame with respect to A frame
ω_0	Angular velocity of A frame with respect to inertial space

Abstract

Equations of motion for a satellite controlled through continuous, low thrust propulsion systems are analyzed through numerical integration techniques. The equations of motion are derived using the Euler Moment Equations. The properties of the satellite model are based upon the Intelsat VII design of communications satellite. A simple rate and error feedback controller is used in providing active attitude control about two and three axis. Perturbation models are created and applied based upon the satellite model. Parameters varied include thrust levels and controller deadband widths. System response times, pointing accuracy, and total impulse required for attitiude control are determined as measurements of relative performance.

APPLICATION OF LOW THRUST PROPULSION TECHNIQUES TO SATELLITE ATTITUDE CONTROL SYSTEMS

I. Introduction

Satellite designers are constantly working towards improving the performance and life expectancy of their spacecraft designs. One approach to this goal has been the use of non-chemical propulsion techniques, such as electrothermal or electrostatic, in the attitude control systems. Some examples of research in this area are presented by Beattie [4], Burton [6,7], Ghislanzoni [10], Hirata [12], Sovey [17], and Valentian [19]. These examples cover just a few of the possible non-chemical propulsion techniques available today or in the near future. Non-chemical methods have the potential for a much higher specific impulse than is available from chemical propulsion, resulting in lower fuel consumption and thus a longer satellite lifetime for a given propellant mass. However, they also require a electrical power supply. If the power requirements are to be kept reasonable for a solar powered satellite, the maximum possible thrust is low.

This low thrust can be advantageous, in that greater control of total impulse applied over a period of time is possible. However, it may also requires a different approach to modeling spacecraft attitude dynamics. Typically, chemical thrusters have a relatively high impulse delivered per

shortest possible firing time. This **minimum impulse bit** limits the maximum possible accuracy of the system. If too narrow of limits are attempted for the attitude accuracy, the thrusters fire nearly continuously, with each pulse pushing the spacecraft axis from one side of the limits to the other. The much smaller minimum impulse bit available to low-thrust propulsion can allow much smaller accuracy limits to be used.

If the primary mode of attitude control is the use of reaction wheels, with the thrusters merely used to remove excess momentum due to secular perturbations, the accuracy of the system is not limited by the minimum impulse bit, as long as the momentum wheels are capable of counteracting the impulse. In this case, determining relative effectiveness of different propulsion systems simply involves calculating the total momentum to be removed over the life expectancy of the satellite and determining the propulsion system that requires the least mass in order to accomplish this. However, if the propulsion system is the primary attitude control actuator, then the effectiveness of the system depends upon the magnitude of the impulse bit and the frequency of thruster operations.

Therefore, this thesis will examine the spacecraft attitude dynamics, using a non-linear numerical analysis scheme. The primary goal is to determine the potential effectiveness of low-level propulsion in controlling the attitude of a satellite, and investigate how the application of these systems to attitude control differs from that of chemical thrusters. Figures of merit will

include the ability to maintain optimal attitude in the presence of perturbative effects, time required to realign the attitude, and the thruster usage percentage.

The analysis will use a non-linear dynamical model of a non-spinning, non-symmetrical satellite in a geosynchronous orbit about the Earth. Perturbative influences examined will include gravitational gradient torques, solar radiation pressure, radio-frequency transmission pressure, and the effects of imperfections in the spacecraft orbit. Perturbations that affect the orbit itself will not be modeled, as station keeping methods using low-thrust propulsion have already been examined in many papers, such those by Day [8], Ghislanzoni [10], and Sovey [17], and are outside of the area of interest in this discussion. Activation of the thrusters will be determined though a simple rate-and-error feedback controller.

In addition to gaining insight into the relative performance of low thrust levels, this analysis will also examine the magnitude of the perturbative effects and the lower limits they set upon attitude control systems. While the perturbations due to solar radiation and radio-frequency pressures are highly dependent upon spacecraft configuration, the other perturbations will be applicable to any spacecraft.

All calculations and simulations will be based upon several assumptions. First, the mass properties of the satellite will be held constant. This means that the mass expelled by the thrusters is considered to be negligible, and that the spacecraft is treated as a rigid body. In addition, all thrusters are assumed to

be perfectly aligned, with repeatable performance.

The information to be gained through this analysis will aid in determining the effectiveness of non-chemical attitude control systems. It will provide figures of merit for use in comparing possible performance with that of chemical systems, and will be of use in the design and selection of satellite systems.

II. Theory

This chapter follows the derivation of the equations of motion of a non-symmetrical, 3-axis stabilized satellite. These equations are derived using Newtonian dynamics, resulting in a set of second order, nonlinear differential equations. A satellite model is then created and used to determine the perturbative forces acting on the satellite. A range of thrust levels of interest is determined, and two simplified controllers are designed using rate and position feedback.

2.1 Equations of Motion

Euler's Moment Equations are used as a starting point in the derivation of the equations of motion for the attitude of a non-symmetric satellite. These equations are reduced to the form of six nonlinear, coupled, first order differential equations to form a state vector for the satellite attitude. The equations are left in a general form to allow for greater flexibility in the analysis. Since numerical integration techniques are used in the analysis, the equations are not linearized. However, the development of these equations follows the linear derivations presented by Agrawal [1:106-131].

2.1.1 Angular Attitude Velocity. The coordinate frames used in this development include a body frame B, a nominal attitude frame A, and an

inertial reference frame I. Figure (2.1) shows the orientation of the nominal attitude frame with respect to the inertial reference frame. The **Z** axis lies in the plane of the orbit and points along the radius vector towards the Earth. The **X** axis lies in the plane of the orbit in the direction of satellite motion. The **Y** axis is perpendicular to the orbit plane, and completes a standard Right-Hand-Set coordinate system.

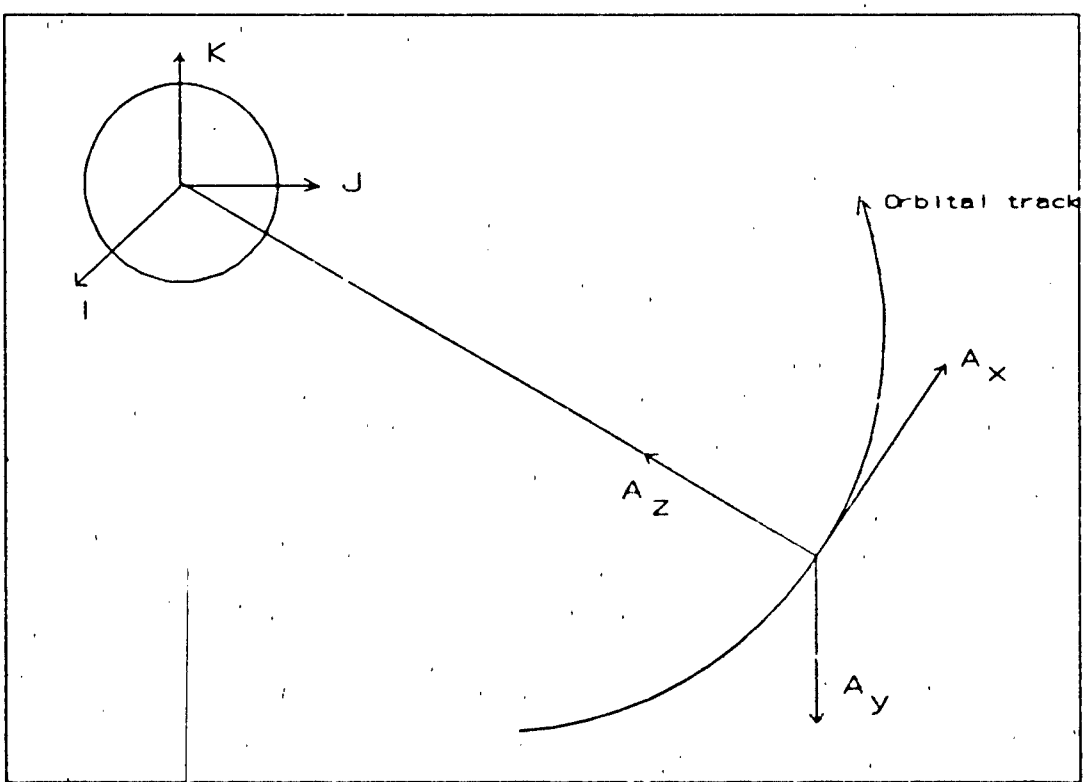

Figure 2.1 Nominal Attitude Coordinate Frame vs. Inertial Frame

The body frame is obtained from the A frame by a 3-2-1 rotation through the angles ψ, θ, and ϕ. The relationship between these two frames is shown in figure (2.2). The body frame is fixed with respect to the satellite and is aligned

along the Principal Moment Axis. Therefore, the satellite's angular velocity is the sum of the rotation rates from the inertial frame to the body frame.

$$\omega^{B/I} = \omega^{B/A} + \omega^{A/I} \qquad (2.1)$$

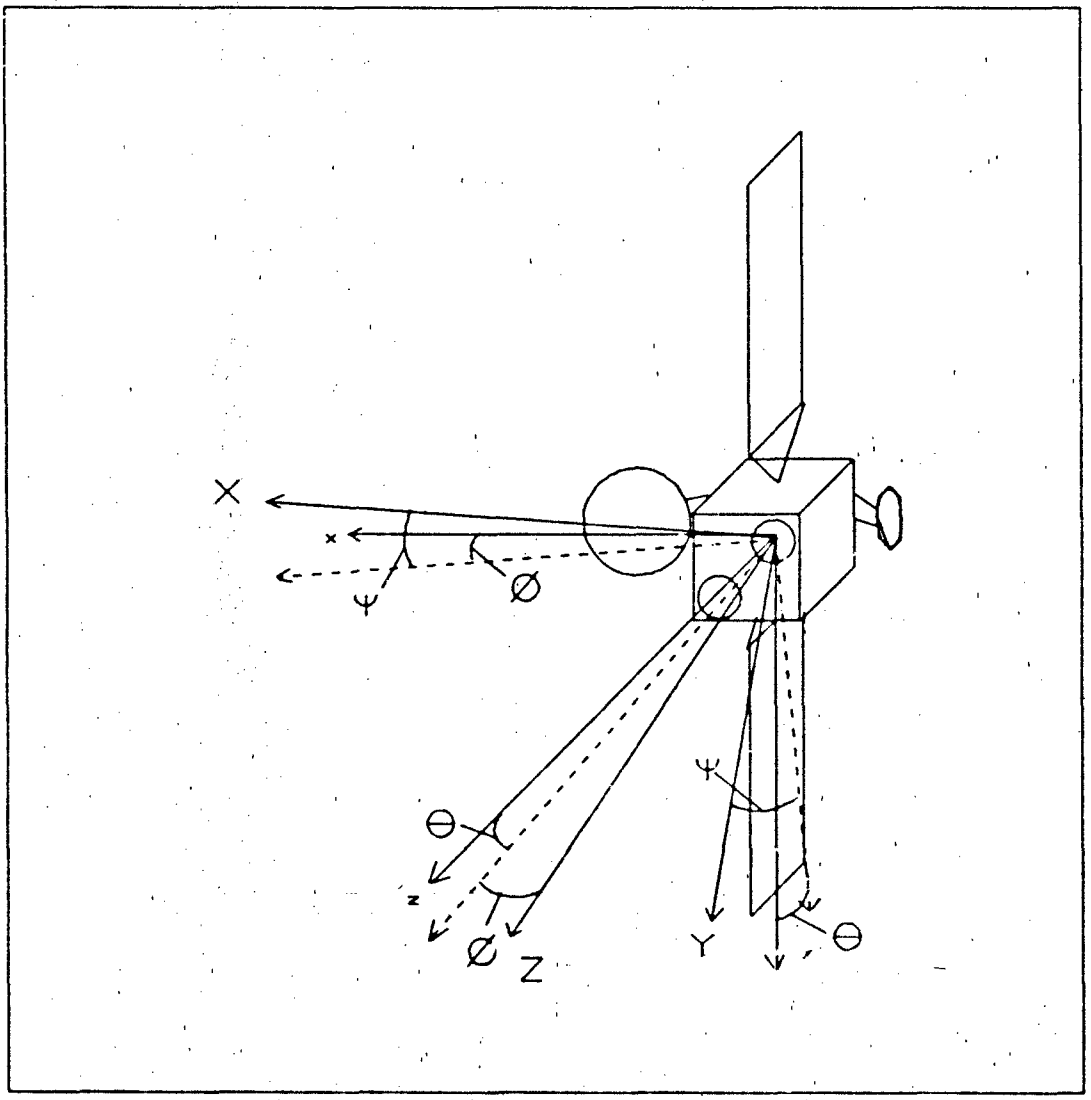

Figure 2.2 Spacecraft Body Frame vs. Nominal Attitude Frame

The rotation rate of the A frame with respect to the inertial frame is the rate of

change of the true anomaly for the orbit, w_0, with the axis of rotation being along the -**Y** axis.

The rotation rate of the body frame with respect to the A frame can be broken down into three parts - the yaw rate $d\psi/dt$ about the **Z** axis, the pitch rate $d\theta/dt$ about an intermediate axis **Y'**, and the roll rate $d\phi/dt$ about the **X"**, or **x** axis. Therefore:

$$\omega^{B/A} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \dot{\phi} + \begin{Bmatrix} 0 \\ \cos\phi \\ -\sin\phi \end{Bmatrix} \dot{\theta} + \begin{Bmatrix} -\sin\theta \\ \cos\theta \sin\phi \\ \cos\theta \cos\phi \end{Bmatrix} \dot{\psi} \qquad (2.2)$$

for use in the analysis, the rotation rates must be expressed in a common reference frame, and the body frame was chosen in order to simplify perturbation modeling and the Moment of Inertia matrix. The rotation matrix from the A to the B frame as the result of the 3-2-1 rotation is:

$$\begin{Bmatrix} i \\ j \\ k \end{Bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix} \begin{Bmatrix} I \\ J \\ K \end{Bmatrix} \qquad (2.3)$$

The intermediate axis system is obtained from the body frame through the rotation through ψ about the **z** axis. Summing these angular velocities results

in the total angular velocity of the body frame with respect to inertial space:

$$\omega^{B/I} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} - \omega_0 \begin{Bmatrix} \cos\vartheta\sin\psi \\ \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi \\ -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \end{Bmatrix} \tag{2.4}$$

To simplify equations, this will be written as

$$\omega = [R]\dot{\varepsilon} - \omega_0 N \tag{2.5}$$

2.1.2 Angular Momentum. Since the angular momentum of the system is the sum of the momenta of its parts, the satellite angular momentum can be divided into that of the rotating body ($\mathbf{H}_b$) and that of any reaction or momentum wheel system ($\mathbf{h}$).

$$H = H_b + h \tag{2.6}$$

where the momentum of the rotating body is

$$H_b = [I]\omega^{B/I} \tag{2.7}$$

The rate of change of the angular momentum of the system equals the external moments applied to that system, and determining the derivative of the

angular momentum vector in a non-inertial frame results in:

$$\boldsymbol{M} = \dot{\boldsymbol{H}} = {}^{B}\dot{\boldsymbol{H}}_{b} + \omega^{B/I} \times \boldsymbol{H}_{b} + {}^{B}\dot{\boldsymbol{h}} + \omega^{B/I} \times \boldsymbol{h} \tag{2.8}$$

Replacing $\mathbf{H}_b$ with equation (2.5) and applying the chain rule for derivatives,

$$\boldsymbol{M} = [I]\omega^{B/I} + [I]\dot{\omega}^{B/I} + \omega^{B/I} \times [I]\omega^{B/I} + {}^{B}\dot{\boldsymbol{h}} + \omega^{B/I} \times \boldsymbol{h} \tag{2.9}$$

Assuming a rigid satellite structure, the moments of inertia are constant and d[**I**]/dt =[0].

The momentum and rate of change of the momentum of any momentum wheel system (**h** and d**h**/dt), if one is used in the spacecraft, depends upon the control system used and its effects upon the wheel speeds. Therefore these two vectors will be left as parameters to be varied as required.

Replacing dω/dt with the derivative of equation (2.5) results in:

$$\boldsymbol{M} = [I]\left[[\dot{R}]\vec{\varepsilon} + [R]\ddot{\vec{\varepsilon}} - \dot{\omega}_0 \boldsymbol{N} - \omega_0 \dot{\boldsymbol{N}}\right] + \omega^{B/I} \times [I]\omega^{B/I} + {}^{B}\dot{\boldsymbol{h}} + \omega^{B/I} \times \boldsymbol{h} \tag{2.10}$$

Solving for $d^2\varepsilon/dt^2$:

$$\ddot{\vec{\varepsilon}} = [R]^{-1}[I]^{-1}\left[\boldsymbol{M} - \omega^{B/I} \times [I]\omega^{B/I} - {}^{B}\dot{\boldsymbol{h}} - \omega^{B/I} \times \boldsymbol{h}\right] - [R]^{-1}\left[[\dot{R}]\vec{\varepsilon} - \dot{\omega}_0 \boldsymbol{N} - \omega_0 \dot{\boldsymbol{N}}\right] \tag{2.11}$$

2.1.3 Reduction to State Vector format. Since the body frame is aligned with the principal axis, [I] is a diagonal matrix and is easily inverted:

$$[I]^{-1} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \tag{2.12}$$

The [R] matrix is more complex, but the inverse can still be determined, along with the first derivative of [R]:

$$[R]^{-1} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \tag{2.13}$$

$$[\dot{R}] = \begin{bmatrix} 0 & 0 & -\dot{\theta}\cos\theta \\ 0 & -\dot{\phi}\sin\phi & -\dot{\theta}\sin\theta\sin\phi + \dot{\phi}\cos\theta\cos\phi \\ 0 & -\dot{\phi}\cos\phi & -\dot{\theta}\sin\theta\cos\phi - \dot{\phi}\cos\theta\sin\phi \end{bmatrix} \tag{2.14}$$

Separating the vectors into their components results in :

$$\begin{Bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{Bmatrix} = [R]^{-1} \begin{Bmatrix} \frac{1}{I_{xx}}(M_x-(I_{zz}-I_{yy})\omega_y\omega_z-\dot{h}_x-h_z\omega_y+h_y\omega_z) \\ \qquad +\dot{\omega}_0 N_x+\omega_0\dot{N}_x+\dot{\psi}\dot{\theta}\cos\theta) \\ \frac{1}{I_{yy}}(M_y-(I_{xx}-I_{zz})\omega_x\omega_z-\dot{h}_y-h_x\omega_z+h_z\omega_x) \\ \quad +\dot{\omega}_0 N_y+\omega_0\dot{N}_y+\dot{\phi}\dot{\theta}\sin\phi+\dot{\theta}\dot{\psi}\sin\theta\sin\phi-\dot{\phi}\dot{\psi}\cos\theta\cos\phi) \\ \frac{1}{I_{zz}}(M_z-(I_{yy}-I_{xx})\omega_x\omega_y-\dot{h}_z-h_y\omega_x+h_x\omega_y) \\ \quad +\dot{\omega}_0 N_z+\omega_0\dot{N}_z+\dot{\phi}\dot{\theta}\cos\theta+\dot{\theta}\dot{\psi}\sin\theta\cos\phi+\dot{\phi}\dot{\psi}\cos\theta\sin\phi) \end{Bmatrix} \quad (2.15)$$

Applying equation (2.13) and simplifying results in three second- order equations:

$$\begin{aligned} \ddot{\phi} = & \frac{1}{I_{xx}}(M_x-(I_{zz}-I_{yy})\omega_y\omega_z-\dot{h}_x-h_z\omega_y+h_y\omega_z) \\ & +\frac{\sin\phi\tan\theta}{I_{yy}}(M_y-(I_{xx}-I_{zz})\omega_x\omega_z-\dot{h}_y-h_x\omega_z+h_z\omega_x) \\ & +\frac{\cos\phi\tan\theta}{I_{zz}}(M_z-(I_{yy}-I_{xx})\omega_x\omega_y-\dot{h}_z-h_y\omega_x+h_x\omega_y) \\ & +\dot{\omega}_0\frac{\sin\phi}{\cos\theta}+\omega_0(\dot{\psi}\frac{\cos\psi}{\cos\theta}+\dot{\phi}\tan\theta\cos\psi)+\frac{\dot{\theta}\dot{\psi}}{\cos\theta}+\dot{\theta}\dot{\phi}\tan\theta \end{aligned} \quad (2.16)$$

$$\begin{aligned} \ddot{\theta} = & \frac{\cos\phi}{I_{yy}}(M_y-(I_{xx}-I_{zz})\omega_x\omega_z-\dot{h}_y-h_x\omega_z+h_z\omega_x) \\ & -\frac{\sin\phi}{I_{zz}}(M_z-(I_{yy}-I_{xx})\omega_x\omega_y-\dot{h}_z-h_y\omega_x+h_x\omega_y) \\ & +\dot{\omega}_0\cos\psi+\omega_0(\dot{\phi}\sin\theta\sin\psi-\dot{\psi}\sin\psi)-\dot{\phi}\dot{\psi}\cos\theta \end{aligned} \quad (2.17)$$

$$\ddot{\psi} = \frac{1}{\cos\theta}\left[\frac{\sin\phi}{I_{yy}}(M_y-(I_{xx}-I_{zz})\omega_x\omega_z-\dot{h}_y-h_x\omega_z+h_z\omega_x)\right.$$
$$+\frac{\cos\phi}{I_{zz}}(M_z-(I_{yy}-I_{xx})\omega_x\omega_y-\dot{h}_z-h_y\omega_x+h_x\omega_y) \qquad (2.18)$$
$$-\dot{\omega}_0\sin\theta\sin\psi+\omega_0(\dot{\psi}\sin\theta\cos\psi-\dot{\phi}\cos\psi+\dot{\theta}\cos\theta\sin\psi)$$
$$\left.+\dot{\theta}\dot{\phi}-\dot{\theta}\dot{\psi}\sin\theta\right]$$

These can then be represented as six, first order equations forming the state vector of the satellite attitude dynamics:

$$
\begin{Bmatrix} \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix}' = \begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dfrac{1}{I_{xx}}(M_x-(I_{zz}-I_{yy})\omega_y\omega_z-\dot{h}_x-h_z\omega_y+h_y\omega_z) \\ \quad +\dfrac{\sin\phi\tan\theta}{I_{yy}}(M_y-(I_{xx}-I_{zz})\omega_x\omega_z-\dot{h}_y-h_x\omega_z+h_z\omega_{xz}) \\ \quad +\dfrac{\cos\phi\tan\theta}{I_{zz}}(M_z-(I_{yy}-I_{xx})\omega_x\omega_y-\dot{h}_z-h_y\omega_x+h_x\omega_y) \\ \quad \dot{\omega}_0\dfrac{\sin\phi}{\cos\theta}+\omega_0(\dot{\psi}\dfrac{\cos\psi}{\cos\theta}+\dot{\phi}\tan\theta\cos\psi)+\dfrac{\dot{\theta}\dot{\psi}}{\cos\theta}+\dot{\theta}\dot{\phi}\tan\theta \\ \dfrac{\cos\phi}{I_{yy}}(M_y-(I_{xx}-I_{zz})\omega_x\omega_z-\dot{h}_y-h_x\omega_z+h_z\omega_x) \\ \quad -\dfrac{\sin\phi}{I_{zz}}(M_z-(I_{yy}-I_{xx})\omega_x\omega_y-\dot{h}_z-h_y\omega_x+h_x\omega_y) \\ \quad +\dot{\omega}_0\cos\psi+\omega_0(\dot{\phi}\sin\theta\sin\psi-\dot{\psi}\sin\psi)-\dot{\phi}\dot{\psi}\cos\theta \\ \dfrac{1}{\cos\theta}\left[\dfrac{\sin\phi}{I_{yy}}(M_y-(I_{xx}-I_{zz})\omega_x\omega_z-\dot{h}_y-h_x\omega_z+h_z\omega_x)\right. \\ \quad +\dfrac{\cos\phi}{I_{zz}}(M_z-(I_{yy}-I_{xx})\omega_x\omega_y-\dot{h}_z-h_y\omega_x+h_x\omega_y) \\ \quad +\dot{\omega}_0\sin\theta\sin\psi+\omega_0(\dot{\psi}\sin\theta\cos\psi-\dot{\phi}\cos\psi \\ \quad \left.+\dot{\theta}\cos\theta\sin\psi)+\dot{\phi}\dot{\theta}+\dot{\theta}\dot{\psi}\sin\theta\right] \end{Bmatrix} \tag{2.19}
$$

2.2 Satellite Model

The spacecraft model is based upon an idealization of the Intelsat VII communication satellite. This satellite was chosen since work has already been conducted on the application of non-impulsive propulsion techniques to orbit maintenance [9]. The spacecraft design was simplified, resulting in the layout shown in figure (2.3). This design can be broken down into just a few simple structures for the purpose of analyzing the dynamics of the vehicle. The dimensions for the spacecraft are available from Neyret [15] and Wilson [20:369]. The values are rounded off for simplicity of use.

The center of mass of the spacecraft is located at the geometric center of the central body. The two solar arrays are positioned along the + and - **y** axis, symmetrically with respect to the satellite center of mass. The feed horn structure is modeled as a homogeneous box centered on the +**z** axis. The two antenna reflectors are treated as flat disks. They are mounted with their centers of mass on the **x** axis, the 2.5 meter diameter transmit antenna along the +**x** and the 1.5 meter receive antenna on the -**x**. Both antenna reflectors are canted at a 30 degree angle from the **x-y** plane.

All components are treated as simple, homogeneous structures in the spacecraft model. From this layout, the approximate values for the Moments of Inertia are determined. Since the spacecraft body frame is aligned along the Principal axis, the Moment of Inertia matrix is a simple diagonal matrix.

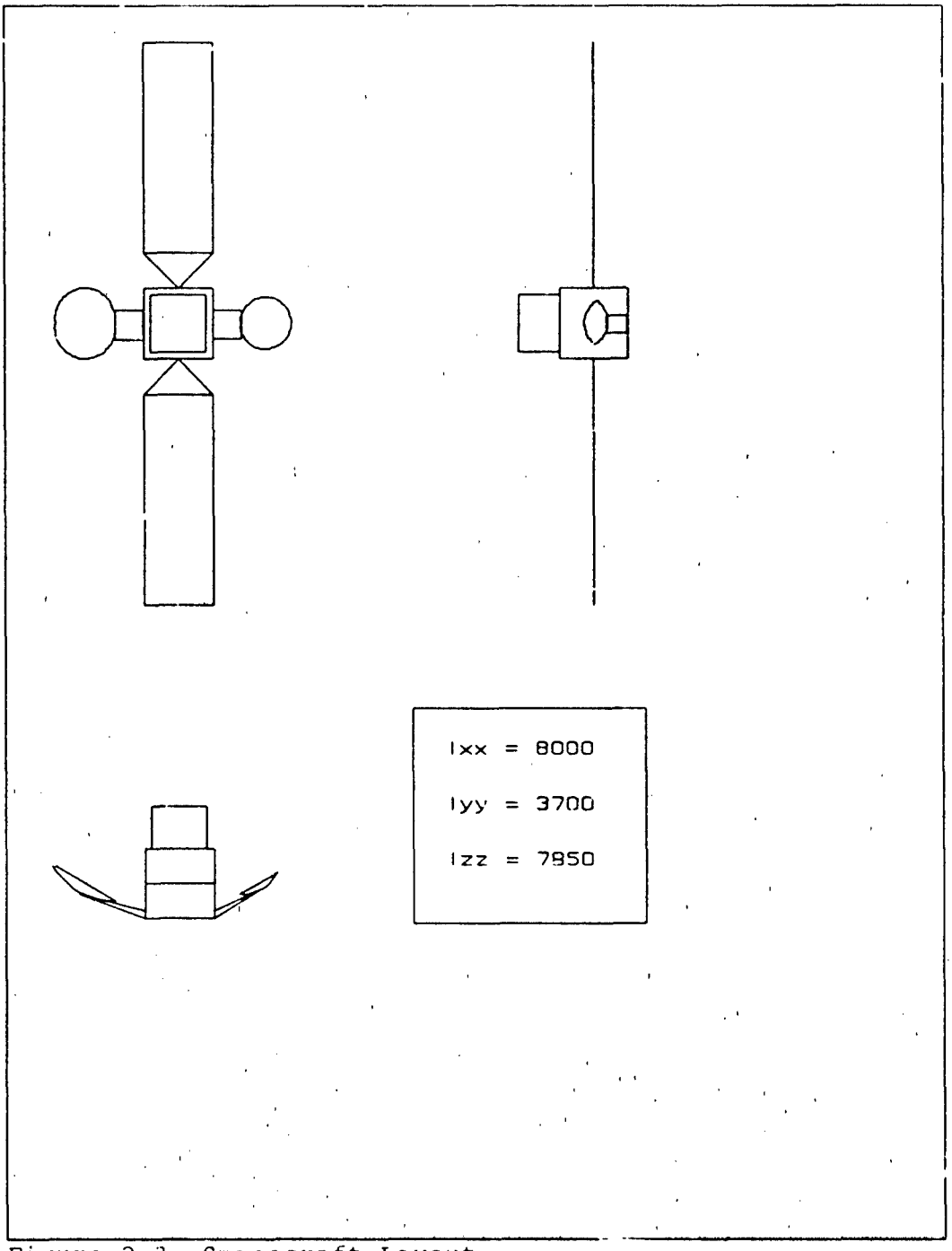

Figure 2.3 Spacecraft Layout

2.3 Perturbative Forces

In the absence of perturbations, it would theoretically be possible to place the satellite into an orbit so that it would maintain a perfect attitude, rotating about its **y** axis once per orbit so as to remain pointing at the same spot on the Earth's surface. In such a case, the only purpose of an attitude control system would be to change the spot on the Earth at which the satellite is pointing. For this purpose, any level of thrust is sufficient, affecting only the time required to change spots. However, the perturbations present in actual satellite operations set lower limits on the thrust levels capable of maintain the desired attitude. In this analysis, a number of sources of perturbations will be considered.

2.3.1 Gravity Gradient Torques. The method followed for determining the gravity gradient torques is based upon the presentation by Agrawal [1:131-133]. The gravitational force, F, operating on a differential element of mass, dM, is:

$$F = \frac{\mu_e R dM}{|R|^3} = \frac{\mu_e (R_0 - r) dM}{|R_0 - r|^3} \qquad (2.20)$$

where **r** is the position vector of the element with respect to the spacecraft center of mass, as shown in figure (2.4).

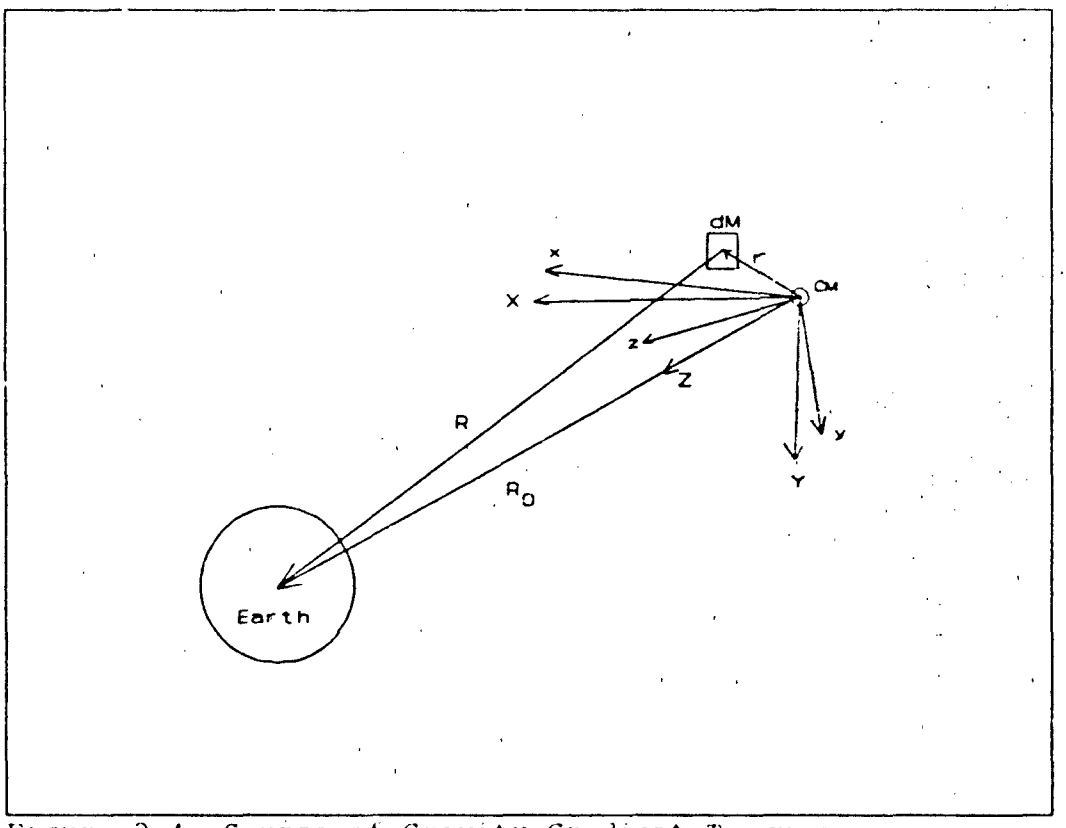

Figure 2.4 Source of Gravity Gradient Torques

Reducing this,

$$\boldsymbol{F} = \frac{\mu_e(\boldsymbol{R}_0 - \boldsymbol{r})dM}{R_0^3}\left[1 - 3\frac{\boldsymbol{r}\cdot\boldsymbol{R}_0}{R_0^2} + o\left(\frac{r^2}{R_0^2}\right)\right]$$

$$\cong \frac{\mu_e(\boldsymbol{R}_0 - \boldsymbol{r})dM}{R_0^3}\left[1 - 3\frac{\boldsymbol{r}\cdot\boldsymbol{R}_0}{R_0^2}\right] \qquad (2.21)$$

The moment on the spacecraft is then the sum of the moments of the differential elements.

$$\boldsymbol{M} = \frac{3\mu_e}{R_0^5} \int (r \times R_0)(r \cdot R_0) dM \tag{2.22}$$

Expressing $\mathbf{R_0}$ in body frame coordinates and separating the moment vector into its components gives:

$$\boldsymbol{M} = \frac{3\mu_e}{R_0^3} \begin{Bmatrix} \int (y^2 - z^2) dM \sin\phi\cos\phi\cos^2\theta \\ \int (x^2 - z^2) dM \sin\theta\cos\theta\cos\phi \\ \int (y^2 - x^2) dM \sin\theta\cos\theta\sin\phi \end{Bmatrix}$$

$$= \frac{3\mu_e}{R_0^3} \begin{Bmatrix} (I_{zz} - I_{yy}) \sin\phi\cos\phi\cos^2\theta \\ (I_{zz} - I_{xx}) \sin\theta\cos\theta\cos\phi \\ (I_{xx} - I_{yy}) \sin\theta\cos\theta\sin\phi \end{Bmatrix} \tag{2.23}$$

Since $I_{xx} > I_{zz} > I_{yy}$, the gravitational moment about the **x** axis is perturbing, and the moment about the **y** axis is correcting. The moment about the **z** axis is a function of the error about that axis only so much as that error is a function of the roll and pitch errors. It is primarily a function of the errors about the **x** and **y** axis.

2.3.2 Solar Radiation Pressure Torques. The reflection or absorption of photons involves a transfer of momentum, resulting in a force acting on the satellite. This force, where non-symmetrical about the center of mass, results in a net torque. Solar radiation pressure acting on a symmetric satellite simply results in a transverse force, with no resulting torque. The calculations used in modeling these forces are similar to those presented in Agrawal [1:133-135].

A percentage of the photons, ρ, will be reflected from the spacecraft, while the remainder will be absorbed In general, the reflected photons would be divided between those diffusely and those specularly reflected, but in order to determine a limiting value for the solar pressure, all reflected photons will be treated as specularly reflected.

The number of photons impinging on a spacecraft surface depends upon the surface's effective area with respect to the sun's radiation. Given a surface of area A, with a normal vector **n**, as shown in figure (2.5), its effective surface area with respect to the direction of the sunlight, **S**, is

$$A_{eff} = A\,|\boldsymbol{n}\cdot\boldsymbol{S}| \qquad (2.24)$$

If a photon is absorbed, it transfers its entire momentum vector to the surface, resulting in a force due to absorption of

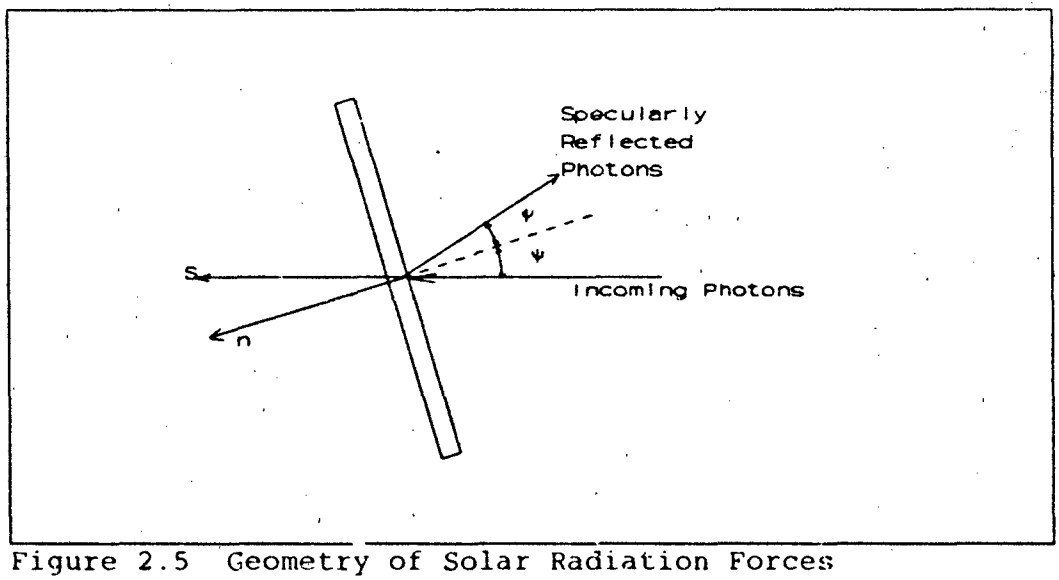

Figure 2.5 Geometry of Solar Radiation Forces

$$F_a = (1-\rho) PA |n \cdot S| S \qquad (2.25)$$

where P is the value of the solar radiation pressure at the satellite's orbit.

For specularly reflected photons, the photons reverse the component of their momentum vector normal to the surface, resulting in a net change of 2p cos Ψ = 2p(n·S), where p is the magnitude of the momentum vector. This change in momentum is applied along the normal vector, **n**, and results in a force due to reflection of:

$$F_r = 2\rho PA |n \cdot S| (n \cdot S) n \qquad (2.26)$$

Combining these two forces gives a net force of:

$$\boldsymbol{F} = PA|\boldsymbol{n}\cdot\boldsymbol{S}|\left((1-\rho)\boldsymbol{S}+2\rho(\boldsymbol{n}\cdot\boldsymbol{S})\boldsymbol{n}\right) \tag{2.27}$$

Therefore the moment acting on the spacecraft due to the solar radiation pressure on this one surface is

$$\boldsymbol{M} = PA|\boldsymbol{n}\cdot\boldsymbol{S}|\boldsymbol{r}\times\left((1-\rho)\boldsymbol{S}+2\rho(\boldsymbol{n}\cdot\boldsymbol{S})\boldsymbol{n}\right) \tag{2.28}$$

where **r** is the vector from the spacecraft center of mass to the center of pressure of the surface. The total moment is then the sum of the moments due to all contributing (non-symmetrical) surfaces.

The satellite model used in this analysis has only three areas of non-symmetry - the feed horn assembly, and the two antenna reflectors. The antenna reflectors are modeled as flat plates, and for the purpose of determining solar pressure, the feed horn assembly is treated as two flat plates perpendicular to each other, one in the **x-z** plane, the other in the **y-z** plane. The areas, position vectors, and normal vectors of these surfaces, expressed in body frame coordinates, are:

Surface	Area	**r**	**n**
2.5 m Antenna	4.91 m^2	(3.40,0,0)	(-0.5,0,0.866)
1.5 m Antenna	1.77 m^2	(-2.85,0,0)	(0.5,0,0.866)
x-z plate	3 m^2	(0,0,2)	(0,1,0)
y-z plate	3 m^2	(0,0,2)	(1,0,0)

Table 2.1 Non-symmetric Components

The unit vector **S** for the direction of the solar radiation is

$$\boldsymbol{S}=\sin\alpha\cos\delta\boldsymbol{I}+\sin\delta\boldsymbol{J}+\cos\alpha\cos\delta\boldsymbol{K} \qquad (2.29)$$

where δ is the declination of the sun and α is the orbit angle as measured from the spacecraft local noon. Translating into body frame coordinates

$$\boldsymbol{S}=\begin{Bmatrix} \cos\theta\cos\psi\sin\alpha\cos\delta+\cos\theta\sin\psi\sin\delta-\sin\theta\cos\alpha\cos\delta \\ (-\cos\phi\sin\psi+\sin\phi\sin\theta\cos\psi)\sin\alpha\cos\delta \\ \quad +(\cos\phi\cos\psi+\sin\phi\sin\theta\sin\psi)\sin\delta+\sin\phi\cos\theta\cos\alpha\cos\delta \\ (\sin\phi\sin\psi+\cos\phi\sin\theta\cos\psi)\sin\alpha\cos\delta \\ \quad +(-\sin\phi\cos\psi+\cos\phi\sin\theta\sin\psi)\sin\delta+\cos\phi\cos\theta\cos\alpha\cos\delta \end{Bmatrix} \qquad (2.30)$$

The position of satellite local noon moves along the orbit at a rate of

approximately 1° per day, but the variation in **S** over one orbit is sufficiently small that **S** can be treated as a constant vector.

Therefore the total torque due to solar radiation pressure is

$$\boldsymbol{M}_s = P \begin{Bmatrix} -2A_3 \left| S_y \right| (1+\rho) S_y - 2A_4 \left| S_x \right| (1-\rho) S_y \\ \\ -3.4A_1 \left| -1/2\, S_x + \sqrt{3}/2\, S_z \right| \left((1+1/2\rho) S_z - \sqrt{3}/2\rho S_x \right) \\ \quad +2.85A_2 \left| 1/2\, S_x + \sqrt{3}/2\, S_z \right| \left((1+1/2\rho) S_z - \sqrt{3}/2\rho S_x \right) \\ \quad -2A_3 \left| S_y \right| (1-\rho) S_x + 2A_4 \left| S_x \right| (1+\rho) S_x \\ \\ 3.4A_1 \left| -1/2\, S_x + \sqrt{3}/2\, S_z \right| (1-\rho) S_y \\ \quad -2.85A_2 \left| 1/2\, S_x + \sqrt{3}/2\, S_z \right| (1-\rho) S_y \end{Bmatrix} \quad \textbf{(2.31)}$$

where S_x, S_y, and S_z are the body frame components of the **S** unit vector.

2.3.3 Transmitter Radio-Frequency Torque. The photons making up the satellite downlink also have momentum, and transmitting them results in a change of momentum to the spacecraft, acting at the transmitting antenna. If the beamwidth of the transmitting antenna is narrow enough, the photons can all be treated as having the same velocity vector. Assuming that this transmitted beam is directed in the **z** direction, the force acting on the antenna is

$$F_t = -\frac{P_t}{c} z \tag{2.32}$$

where P_t is the power transmitted.

The center of pressure of this force is at the geometric center of the antenna, and it has a position vector of

$$r = \begin{Bmatrix} 3.4 \\ 0 \\ 0 \end{Bmatrix} \tag{2.33}$$

Thus the net torque due to Radio Frequency transmission from the primary antenna is

$$M = r \times F = \begin{Bmatrix} 0 \\ 3.4\frac{P_t}{c} \\ 0 \end{Bmatrix} \tag{2.34}$$

A total transmitted power for all channels of 500 watts is assumed for this analysis, based upon data on the capabilities of Intelsat VII presented by Nabi [13], giving a torque of

$$M = 5.66\times10^{-6}\ N\cdot m\ j \tag{2.35}$$

2.3.4 Effects of Imperfect Orbits. If the satellite is in a perfect geosynchronous orbit, with both the eccentricity and the inclination equal to zero, then by rotating the spacecraft at a rate of ω_0 about the **-J** axis, the satellite remains pointing at the same point on the surface of the Earth. However, imperfections in the orbit will have two types of effects and may require the attitude control system to compensate.

The first effect is caused by the fact that the angular motion of the satellite, dv/dt, is not constant in an elliptical orbit. Thus the satellite must change its rate of rotation in order to remain pointing at the center of the Earth. In the derivation of the equations of motion in section (2.1), this rate of rotation corresponds to the angular rate of the A frame with respect to inertial space, and is termed ω_0. Using the relationship between the true anomaly, ν, and the mean motion, shown in Bate [3:185-187], an equation can be derived expressing the rate at which ω_0 changes as a function of ν.

$$\dot{\omega}_0 = \frac{-2n^2 e \sin\nu (1+e\cos\nu)^3}{(1-e^2)^{3/2}} \qquad (2.36)$$

This value can then be used in the state vector equations in order to take these effects into account.

The second effect of a non-geosynchronous orbit is due to the fact that the satellite is often aligned with a specific point on the Earth's surface, and not at the center of the Earth. The vector from the satellite to this surface point is

at an angle to the satellite's position vector with respect to the center of the Earth, as shown in figure (2.6)

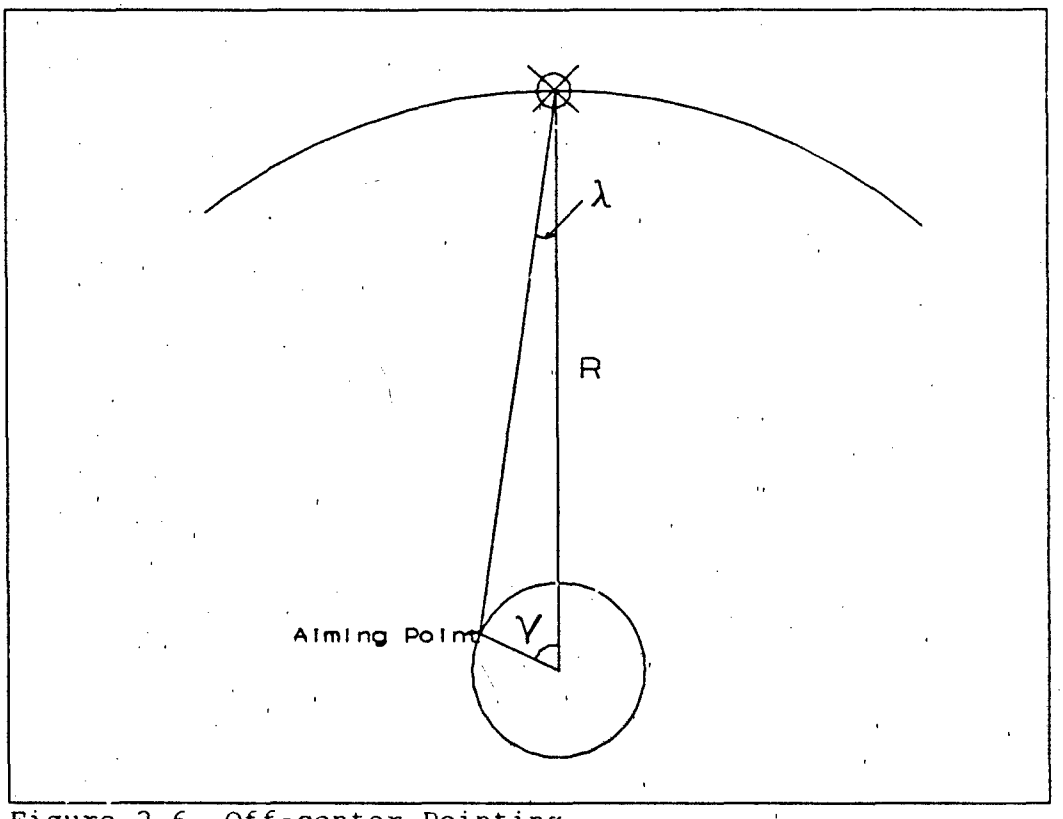

Figure 2.6 Off-center Pointing

The angle γ is the difference between the satellite's position vector and the sub-point position vector with respect to the center of the Earth. Solving for the angle λ results in

$$\cos\lambda = \frac{R-\cos\gamma}{(R^2+1-2R\cos\gamma)^{1/2}} \qquad (2.37)$$

where R is expressed in units of Earth radii.

If the satellite's orbit is circular, but inclined, then γ is a function of the orbital inclination and the angle, u, between the ascending node and the

satellite's current position, and measures the angle from the orbital plane. In this case,

$$\gamma = \sin^{-1}(\sin i \sin u) \tag{2.38}$$

If the satellite is in an elliptical, eccentric orbit, then γ is a function of the true anomaly, ν, and the Earth's rotation since the satellite's last perigee passage. In this case, λ is in the plane of the orbit and has a value of

$$\cos\lambda = \frac{R - \cos(\omega_e t - \nu)}{(R^2 + 1 - 2R\cos(\omega_e t - \nu))^{1/2}} \tag{2.39}$$

If the orbit is both inclined and elliptical, λ is a function of a combination of the two factors. For orbits that are nearly geosynchronous, with inclination less than 2° and eccentricity of less than 0.01, the maximum value for λ is still small, on the order of 0.4°. The second derivative of λ is the value of greatest interest, since it determines the angular acceleration that the satellite must achieve in order to maintain the proper attitude. This can be determined from

$$\ddot{\lambda} = \frac{\ddot{\phi}(R^2 + 1 - 2R\cos\gamma)(R\cos\gamma - 1) - \dot{\phi}^2 R\sin\gamma(R^2 - 1)}{(R^2 + 1 - 2R\cos\gamma)^2} \tag{2.40}$$

where γ is the result of the combined angles determined in equations (2.37) and (2.39).

For an orbit within the limits expressed above, the required angular

acceleration is on the order of 10^{-10} radians/sec^2. For the satellite modelled in this analysis, this requires a torque on the order of 10^{-6} Nm. Note that for satellites in a lower altitude, highly eccentric orbit, such as a Molniya orbit, the requirements for angular acceleration will be much higher.

2.3.5 Net Perturbative Torques. The net torques applied to the satellite are therefore a combination of secular and cyclic components. The major component is caused by the solar radiation pressure, which acts about all three axis if the solar declination is non-zero. In the case of an orbit with a declination of 0°, the solar radiation pressure contributes a moment about only the **y** axis. The torques caused by solar radiation and radio frequency pressure are plotted over the period of one orbit in figures (2.7 - 2.10).

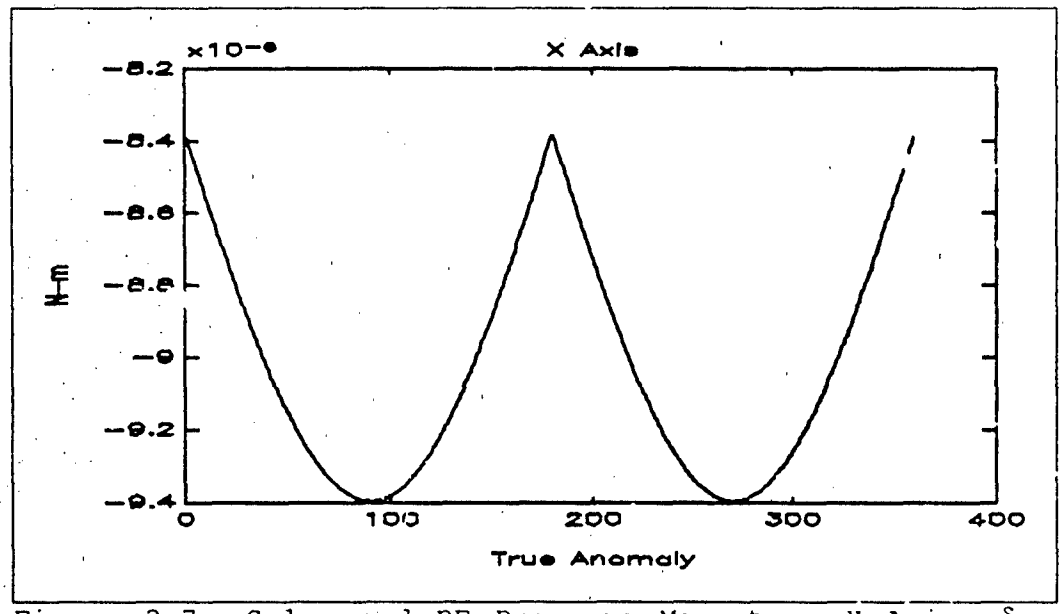

Figure 2.7 Solar and RF Pressure Moments - X Axis, δ = 23.44°, e = 0

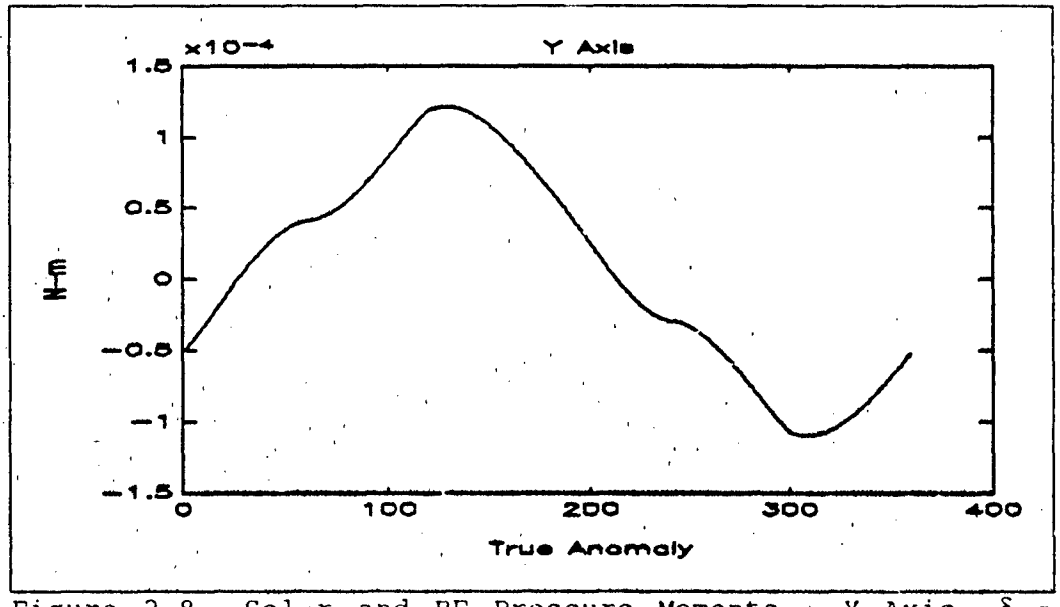

Figure 2.8 Solar and RF Pressure Moments - Y Axis, δ = 23.44°, e = 0

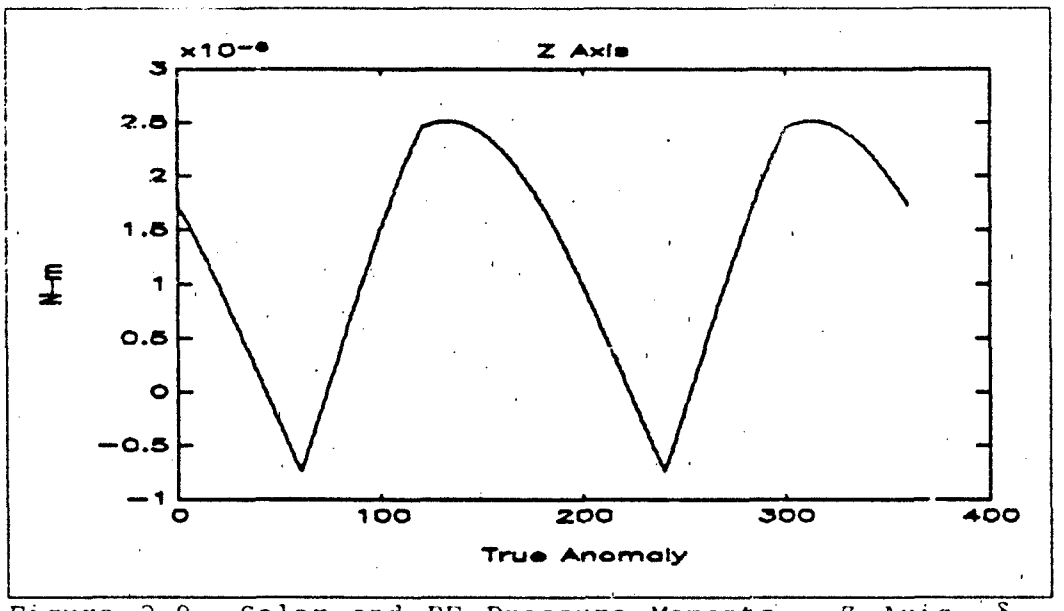

Figure 2.9 Solar and RF Pressure Moments - Z Axis, δ = 23.44°, e = 0

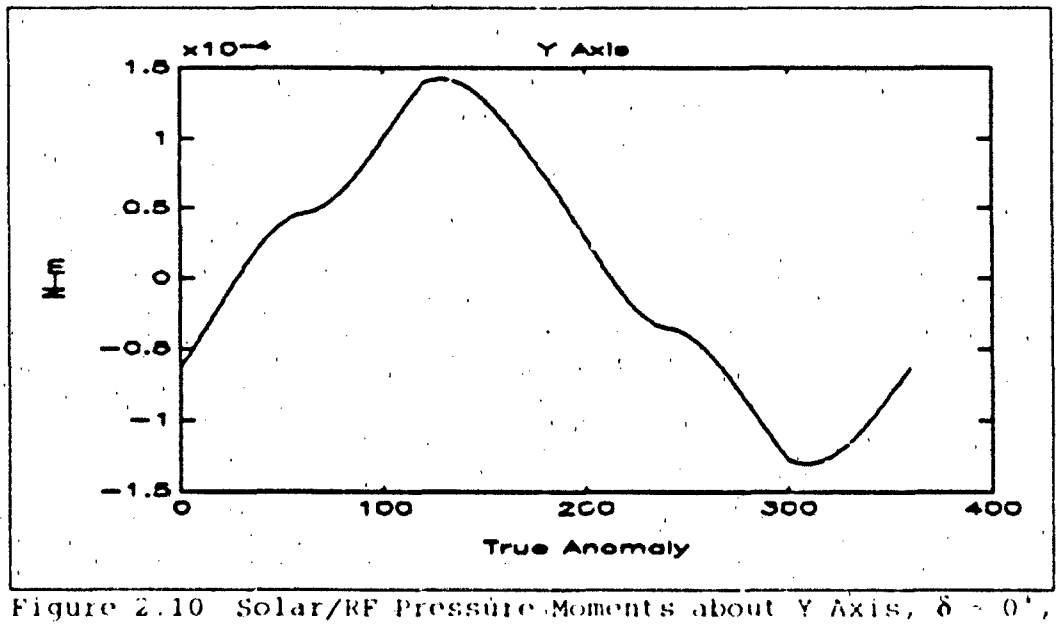

Figure 2.10 Solar/RF Pressure Moments about Y Axis, δ = 0°, e = 0

2.4 Thruster Performance Limits

The thrust of an electric propulsion system depends upon three parameters - the effective exhaust velocity (V_e), the input power (P), and the system efficiency (η). From these three parameters, the thrust delivered can be modeled as

$$Thrust = \frac{2P\eta}{V_e} \qquad (2.41)$$

Current electric propulsion methods have V_e's that range from 3000 m/s for Resistojets to 100000 m/s for Ion engines [16]. Higher levels are possible, but the power levels required for a meaningful level of thrust become exorbitant.

Since the thrusters will be used at short intervals with a somewhat irregular period, they need to be ready to operate at all times. In some forms of electric propulsion, this requires electrical power to maintain the system in a standby mode due to the need to maintain heater temperatures or avoid a length power-up [9:7, 14:7]. Therefore, the use of batteries as the primary source of power is not acceptable, since they would be in constant use. Either additional solar arrays must be added to increase the available power, or the thrusters must operate on the margin of power available after spacecraft payload, housekeeping, and battery recharging demands are met.

The specific mass of the solar arrays and required support equipment for

Intelsat VII is 78 kg/kW [8:5]. Any additional solar arrays added will also increase the satellites moments of inertia.

If no additional solar arrays are added, then the system must require less power than remains after other spacecraft needs are met. On the Intelsat VII, this is 399 watts at End-of-Life power levels [15:4].

System efficiencies vary considerably depending upon the method of propulsion in use. Electrothermal systems have efficiencies up to $\eta = 0.8$, dropping off with increasing V_e to a value of 0.3 at V_e of 10000 m/s. Electrostatic systems also have efficiencies of 0.3 at 10000 m/s, but the efficiencies rise to about 0.8 above V_e of 40000 m/s [6:4].

Using these limits, for a system with V_e of 3000 m/s, P of 400 watts, and η of 0.8, a maximum thrust of 0.22 N is calculated. A lower level of thrust corresponding to a system with lower power or a higher V_e could be used, with the only limit being that the thrust be required to exceed the perturbative forces.

There are far too many types of electrothermal, electromagnetic, and electrostatic propulsion systems for them to be described here. Some, such as resistojets, operate in much the same manner as chemical thrusters, being controlled by limiting the propellant flow [2:2]. Others, such as Radio-frequency ion thrusters, can be controlled by varying the radio power applied to the ionization chamber or by controlling the voltage applied across the accelerator grid [11, 5:2]. Pulsed ion and electrothermal thrusters typically have no direct control of the thrust delivered per pulse. Instead, they control total impulse by

varying the number of pulses used or the pulse duration [7:142, 10:3, 18:1].

2.5 Controller Design

Two different controller designs are used in this analysis. Both are based using upon a rate-and-error feedback system to determine when to fire the thrusters. The first design drives three sets of thrusters, one set about each of the body axis. The other combines a momentum wheel with its spin axis aligned along the **-y** axis and one set of thrusters acting about an axis lying in the **x-z** plane. The systems are based upon calculations provided by Agrawal [1:137-146] and Dougherty [9].

In the controller developed to command the three-axis thruster package, a linear, non-coupled set of equations is used so that each axis may be considered separately. The system uses rate and error feedback to develop a proportional signal for the thrusters. In chemical thrusters, such a signal is typically used to modulate the pulsing of the thruster valves, thus producing a net impulse proportional to the signal [1:142]. The signal is determined by

$$M_c = K_a(\tau\dot{\theta} + \theta) \tag{2.42}$$

where

$$K_a = \frac{Thrust \cdot Moment\ arm}{deadband} \tag{2.43}$$

$$\tau = 2\sqrt{\frac{I_a}{K_a}} \qquad (2.44)$$

This produces a signal profile as shown in figure (2.11). However, since the thruster are being treated as on/off devices, with each thruster pulse lasting for one time step, the impulse delivered to the spacecraft is as shown in figure (2.12).

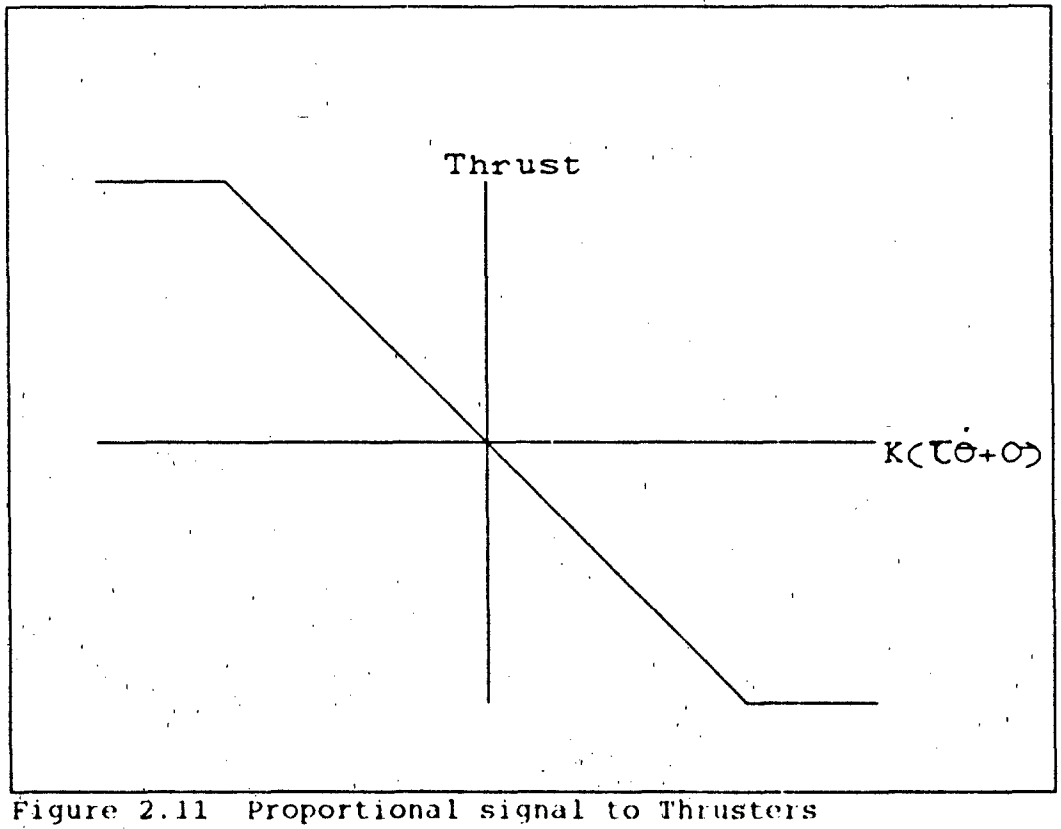

Figure 2.11 Proportional signal to Thrusters

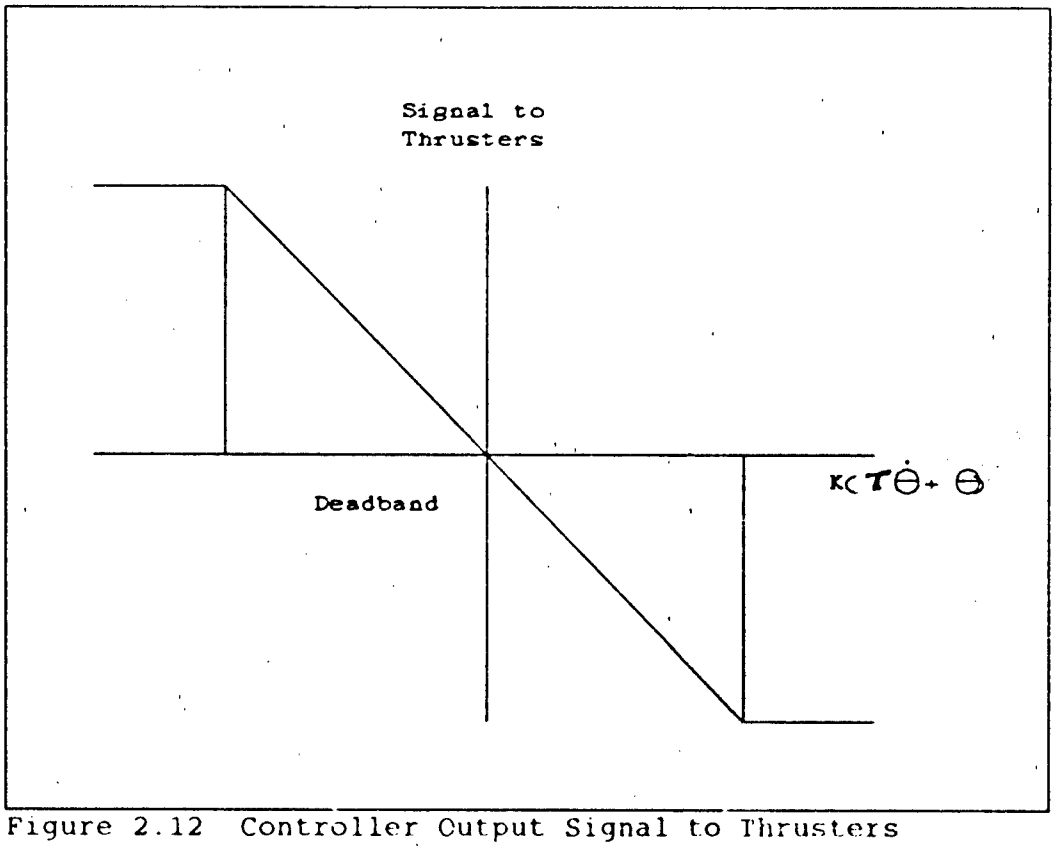

Figure 2.12 Controller Output Signal to Thrusters

The second method investigated in this thesis actively controls the pitch and roll axis, while leaving the yaw axis to be controlled through gyroscopic coupling. This is done because of the difficulty and expense in accurately sensing yaw errors throughout the entire orbit [1:142]. While sun sensors can be used, they do not function during eclipse season at either satellite noon or midnight.

The errors about the **y** axis are controlled through a reaction wheel, with its angular momentum vector aligned along the -**y** axis of the spacecraft. By controlling the speed of the reaction wheel, the angular velocity of the

spacecraft about that axis is controlled. The wheel is biased so as to always have angular momentum, thus providing gyroscopic stiffness to the satellite.

Because of this gyroscopic effect, the **x** and **z** axis are coupled, and attitude errors in both can be corrected with a single set of thrusters. The sensors detect only the roll errors, but as the satellite proceeds around its orbit, the yaw and roll errors interchange, upon which they are detected and corrected. The thrusters pairs are not mounted directly along the spacecraft **y-z** plane, but are instead offset so as to provide torque about both the roll and the yaw axis. This offset is selected so as to damp out the oscillations introduced by the spacecraft's orbital frequency, and is calculated from

$$\zeta = \tan^{-1} 2\sqrt{\frac{I_{zz}\,\omega_0}{N_m h}} \tag{2.45}$$

where h is the magnitude of the momentum wheel's angular momentum and

$$N_m = \frac{1}{1+h^2/(I_{zz}\,K\cos\zeta)} \tag{2.46}$$

The profile for this system is identical in shape to that of the previous method. The equations used in the controller are

$$M_x = -K\cos\zeta(\tau\dot{\phi}+\phi) \tag{2.47}$$

$$M_z = K \sin\zeta (\tau\dot{\phi} + \phi) . \tag{2.48}$$

where

$$\tau = 2\sqrt{\frac{I_{xx}}{N_m K \cos\zeta}} \tag{2.49}$$

These two control systems will both be used and the results compared in order provide repeatability. This will assure that any characteristics of operation revealed are not caused solely by the controller used in the study.

III. Method of Numerical Analysis

This chapter provides a description of the numerical integration techniques used in this analysis. The Haming integrator, the method used in propagating the state vector, will be described, along with the requirements for its use. The characteristics of the system modeled will then be specified. The parameters to be varied will be described, and the range of values chosen for the investigation will be defined. Some of the difficulties encountered in using the Haming to model this system will be discussed. Finally, the programs developed and their use will be described.

The dynamical system developed in this analysis was encoded using Fortran 77, and compiled and executed on the AFIT computer network Sun workstations. One form of the code developed is presented in Appendix A. Several other forms of the code were created using different initial condition, output, and the attitude controller sections. All versions use identical spacecraft equations of motion and perturbation equations.

3.1 The Haming Integrator

The Haming Integrator uses a fourth-order prediction-correction algorithm in order to integrate first-order differential equations, such as the equations of motion of the satellite attitude state vector presented in equation (2.19). It requires continuous differential equations, with continuous derivatives of those

equations. If an equation has a discontinuity, Haming treats it as a continuous equation through a form of curve-fitting. In order to integrate discontinuous equations, the state vector must be propagated across the discontinuity using some other method, and Haming reinitialized.

Haming also requires that the differential equations be smooth over the interval determined by the step size. If a differential equation varies too greatly over one step, the Haming integrator algorithm used fails to initialize. Reducing the step size can solve this problem, but the integrations take a correspondingly larger amount of processing capability.

3.2 Special Requirements for Modeling with Haming

The largest problem encountered in propagating the state vector with a Haming integrator is the need for a very small step size, especially as the desired attitude error limits get smaller. This requirement is most obvious when the thrusters are active. By reducing the step size, the processing time required is increased. Since the angular accelerations due to the perturbations are very small, the state vector must be propagated through a relatively long period in order to collect the desired data. Thus, the use of Haming and a computer model to examine this subject is very expensive in terms of computing power.

Discontinuities in the equations of motion also require special care. Activating and deactivating thrusters results in a discontinuity in the applied

moments. As a result, the iteration must be halted and Haming reinitialized with the new moments applied. By treating the activation/deactivation of thrusters as an instantaneous event, with no rise or fall time in the thrust, no other method of propagation is required, since the new initial state vector is simply the same as the last state vector obtained. For long thrust times this is an accurate assumption. For burn times that are short relative to the rise and fall times, this method may still be used if the effective thrust used reflects an average thrust over the burn time.

Discontinuities also occur in the moments due to the solar radiation pressure. These are spaced around the orbit at approximately 60° inlervals, and are a result of the different sections rotating to an edge-on aspect to the sun, resulting in a sharp minimum in solar pressure at that attitude. The exact time depends upon the satellite's attitude and therefore must be determined as the state vector is propagated. At each of these discontinuities, the iteration is halted, and the state vector is propagated across the discontinuity by

$$\dot{X}(t+\Delta t) = X(t)+\Delta t \cdot F(t) \qquad (3.1)$$

Haming is then reinitialized and the process continues. The loss of accuracy through using this form for the propagation is minimal due to the small step size and the small changes in the derivatives of the state vector over one step.

These breaks in the propagation are used to allow greater efficiency in the program. During the periods the thrusters are inactive and the rate of

change of the state vector is small, the step size is increased, and processing is faster. When the thrusters are active, Haming is reinitialized with a smaller step size to allow the algorithm to integrate the equations of motion more accurately. Thus the program matches the step size to the requirements of the moment.

3.3 Description of Programs

One principal program was written for this analysis, and several variants created from it in order to investigate different facets. The primary program includes the assignment of constants, the equations for the perturbations, the first-order derivatives of the state vector elements for use by Haming, Haming itself, and an iterative loop to initialize and then drive Haming though the desired interval. The variants added the desired controller and its implementation, output, and initial condition sections, and made necessary changes to the iterative loop.

The form of the code included in this paper propagates the equations of motion throughout one orbit of the satellite about the Earth. During this time a record is kept of the success of the satellite in maintaining its attitude within the desired limits and the amount of time that the thrusters were used in maintaining that attitude. This information is collected for a range of thrust levels for each of several pointing error deadbands.

The second primary form of the code was developed in order to model

error correction response times. The attitude initial conditions are chosen with a pointing error that is greater than the allowed limit. The program then propagates the state vector until the attitude is corrected, while recording the time required to meet pointing requirements and the total thruster use during that time.

A third form of the program was developed in order to examine the effects of faster controller rates and better control of thruster output. The controller rate is simulated by varying the integration step size, thus performing the controller calculations more often. The control of thruster output is modeled by limiting the number of possible levels the thrusters at which the thrusters may be operated.

In addition, the basic program is used to verify controller operation in the absence of perturbations and to examine the magnitudes of the perturbations and their dependency on the satellite attitude. The program is also used to integrate the perturbative moments and thus determine total impulse required over one orbit.

3.4 System Specifications

The satellite model described in section (2.2) provides the dynamical constants and perturbation equations used in the analysis. The perturbation calculations are performed with the following constants and orbital data:

- ρ=0.9 : Reflectivity of polished aluminum

- δ=23.44° : Maximum declination attainable

- e=0.01 : High end of eccentricity for geostationary satellites

- i=0° : Inclination effects are described in section (2.3.4)

In all cases it is assumed that the desired satellite orientation is with the body frame and the A frame aligned (i.e. antenna directed at the center of the Earth). Each set of integrations begin at satellite noon, which also corresponds to the perigee passage.

3.5 Parameters

The initial data is collected by varying the thrust produced and determining the relative pointing accuracy and the thruster usage required. Thrust is varied using a logarithmic selection of values from a high of 0.2 N to a low of 0.00025 N. The high limit is slightly less than twice the expected maximum as calculated in section (2.4) if only one pair of thrusters are used on the spacecraft. The lower limit is selected so as to provide a control moment that is still greater than the maximum moment due to perturbative forces. The values selected to cover this range are shown in table (3.1).

In addition, the desired pointing error limits are varied to examine the effect that this will have on the performance at different thrust levels. One advantage of low thrust engines is the fact that the total impulse delivered during one thruster action is more controllable than that of a chemical

propulsion system, thus allowing greater pointing accuracy. The pointing error limits examined range between 0.5° and 0.0005°, with the points selected presented in table (3.1).

Finally, the step size used in the Haming integration will be varied. Since the program requires that a thruster pulse last an entire step, this models the effect of different pulse lengths and controller operation rates. The step sizes used are between 0.5/0.1 seconds, which is the largest steps that Haming can complete a simulation with, and 0.05/0.01 seconds. The first of each pair is the step size used when no thrusters are firing, and program changes to the shorter step size when Haming is reinitialized when a thruster is turned on. The smallest pair of step sizes nears the lower practical limit for the computer system used. At this step size, integrating over an entire orbit requires approximately an hour per thruster setting. To complete an entire data collection run with ten thruster levels and ten error limits at this rate would require on the order of eighty hours of computer operations, plus additional time for formatting the results.

The simulations that are processed with the largest step size cover the period of one orbit. The data collected at shorter step sizes represent results over a four hour period of the orbit. In addition, integrations over smaller periods of time are made in order to validate the operation of the model. Other values are also used for these parameters as the data collected shows areas of further interest. These values are specified in Chapter IV as the results are

discussed.

Thrust Levels	Deadbands	Step Size
0.2 N	0.5°	0.5/0.1 sec
0.1 N	0.2°	0.25/0.05 sec
0.05 N	0.1°	0.1/0.02 sec
0.02 N	0.05°	0.05/0.01 sec
0.01 N	0.02°	
0.005 N	0.01°	
0.002 N	0.005°	
0.001 N	0.002°	
0.0005 N	0.001°	
0.00025 N	0.0005°	

Table 3.1 Parameters

IV. Results

The numerical integrations programs are used to generate data on the performance of low-level thrusters over a variety of conditions. The first series of calculations are performed in order to validate the controller design and the satellite dynamics package. Data is then collected on the ability of the system to maintain Earth pointing within the limits specified, and on the thruster usage required in accomplishing this. This data is then examined, and additional calculations performed to more closely investigate areas of interest.

4.1 Verification of Controller

In order to verify the performance of the controllers, an initial computer run is made and the errors graphed versus time. The graphs are made with an initial deadband of 0.5°, step sizes of 0.5/0.1 seconds, and thrust of 0.2 Newtons per thruster. Figures (4.1 - 4) demonstrate the errors for the first controller design. The first two curves resemble the responses expected from an impulsive control scheme, with sharp corners as the pointing errors approach the deadbands and relatively constant rates of change between thruster firings. This is due in part to the step size, which places a lower limit on the minimum impulse allowed. Thus the thrusters are firing for longer than is required, resulting in the cycling between limits seen in the plots.

Figure (4.3) represents a case in which the perturbative forces are acting to keep the error near one limit. This is shown in greater detail in figure (4.4).

The initial portion of each plot also show the effects of the perturbative moments quite clearly.

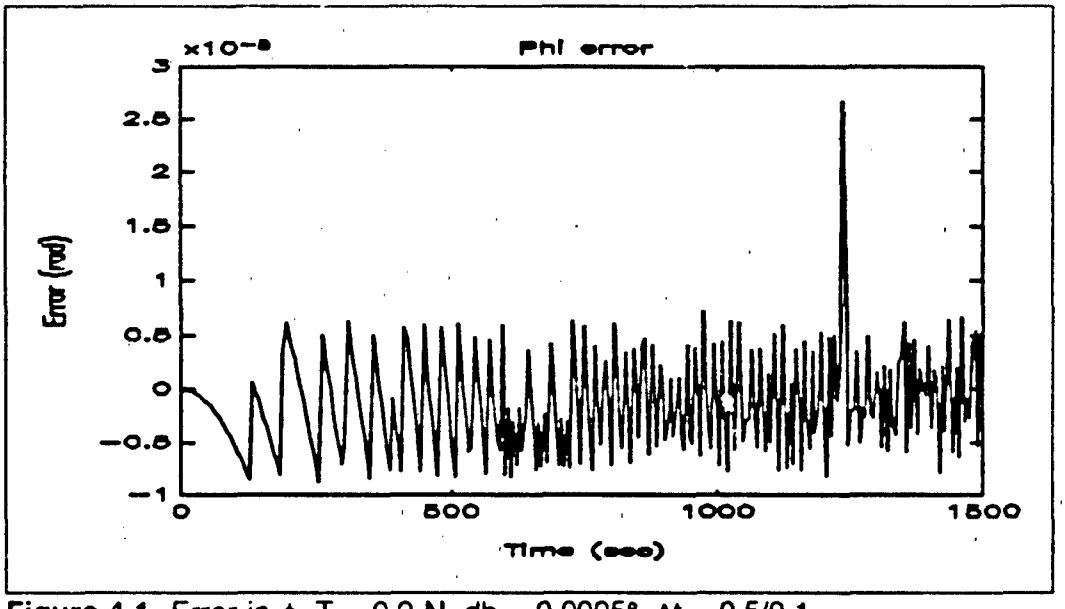

Figure 4.1 Error in ϕ, T = 0.2 N, db = 0.0005°, Δt = 0.5/0.1

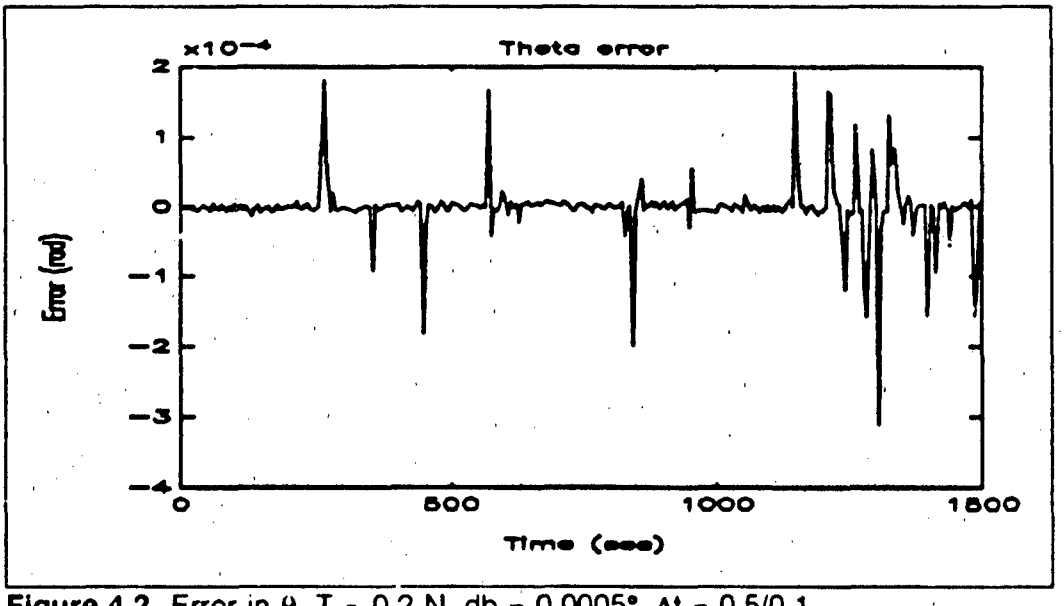

Figure 4.2 Error in θ, T = 0.2 N, db = 0.0005°, Δt = 0.5/0.1

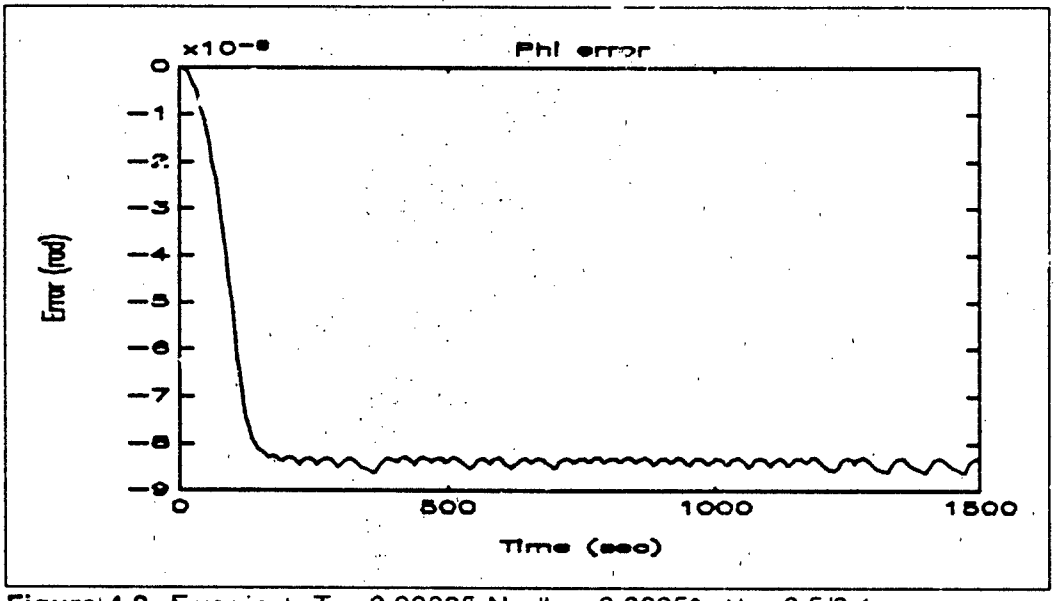

Figure 4.3 Error in ϕ, T = 0.00025 N, db = 0.0005°, Δt = 0.5/0.1

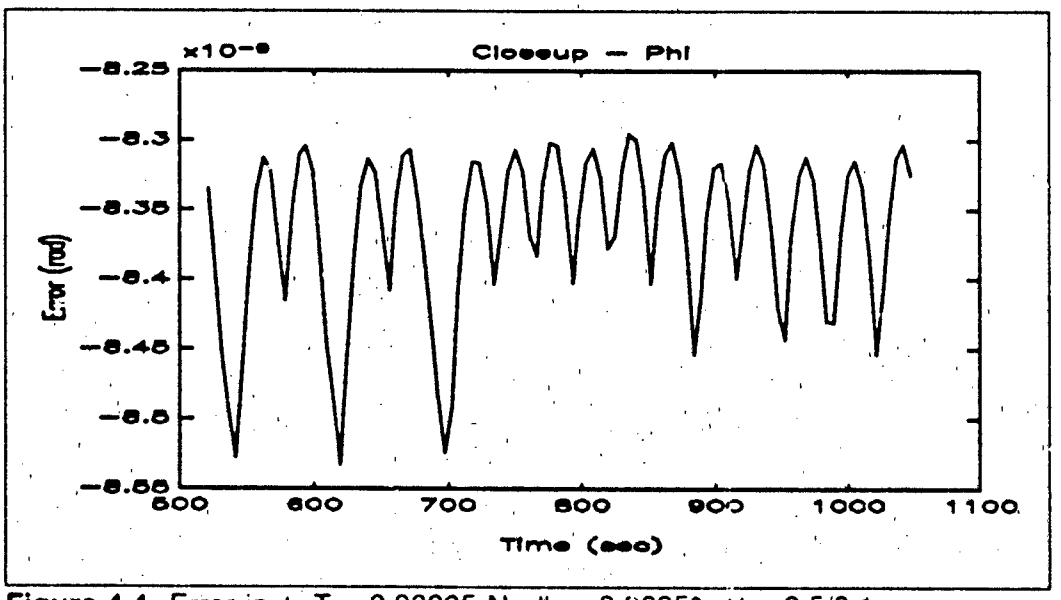

Figure 4.4 Error in ϕ, T = 0.00025 N, db = 0.0005°, Δt = 0.5/0.1

4.2 Pointing Accuracy

The pointing accuracy graphs are constructed by comparing the total time that the error about at least one of axis is greater than the allowable limit and comparing this time with the duration of the integration. These graphs clearly show the accuracy in maintaining narrow pointing limits using a simple on-off controller. The most interesting point about these graphs is the fact that at the narrowest tolerance examined, the more powerful thrusters start to be the least capable. The least powerful thrusters maintain their performance capability despite the change in the limits. The graphs shown here are all using the three-axis control method.

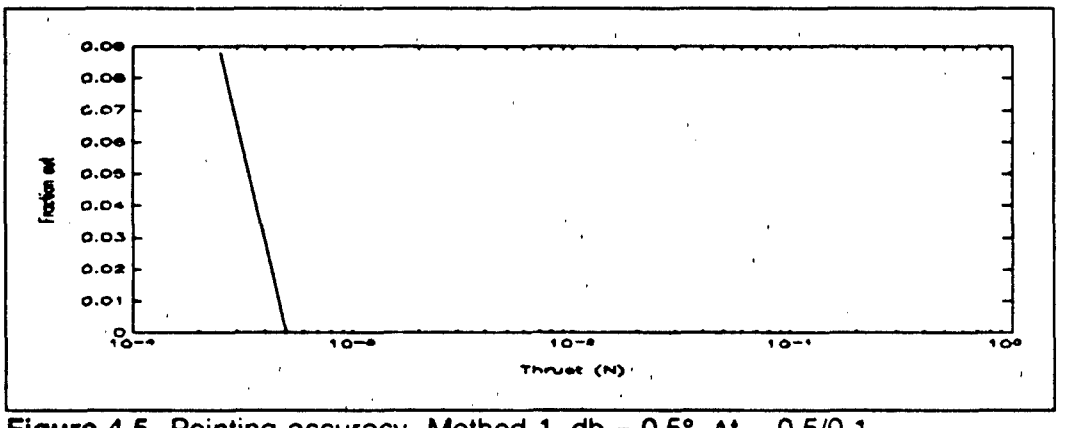

Figure 4.5 Pointing accuracy, Method 1, db = 0.5°, Δt = 0.5/0.1

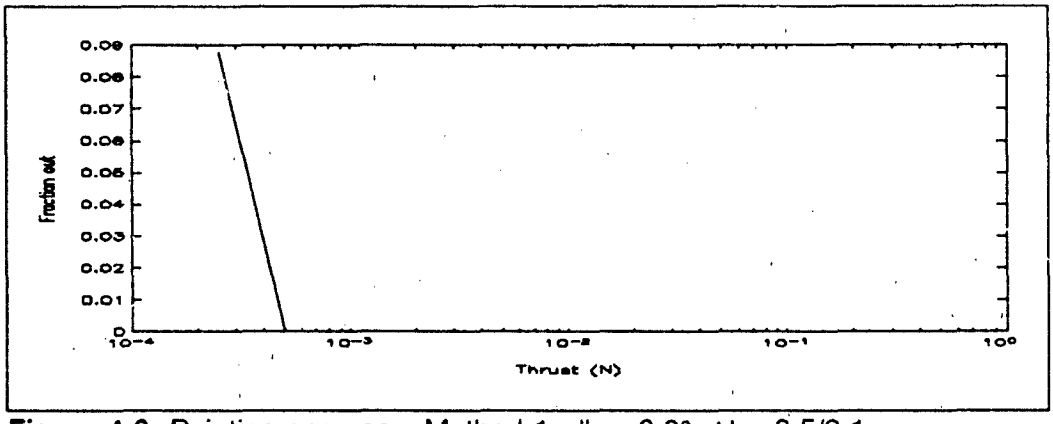

Figure 4.6 Pointing accuracy, Method 1, db = 0.2°, Δt = 0.5/0.1

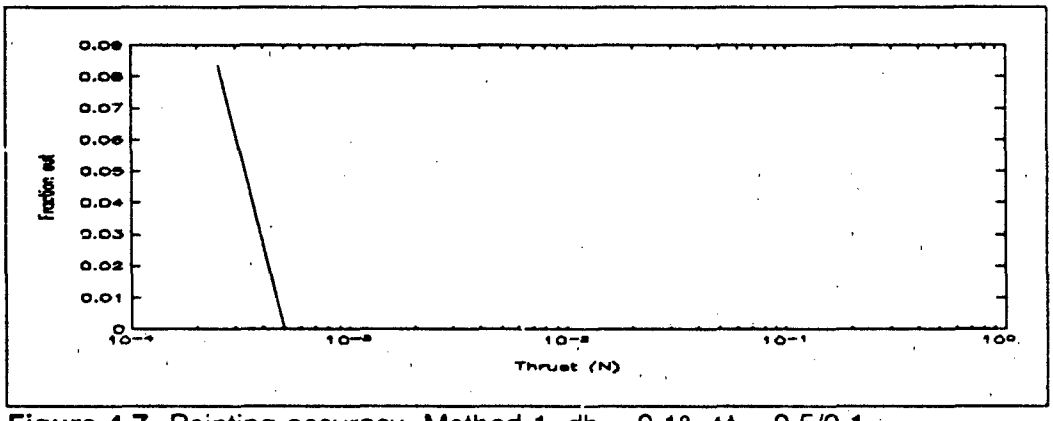

Figure 4.7 Pointing accuracy, Method 1, db = 0.1°, Δt = 0.5/0.1

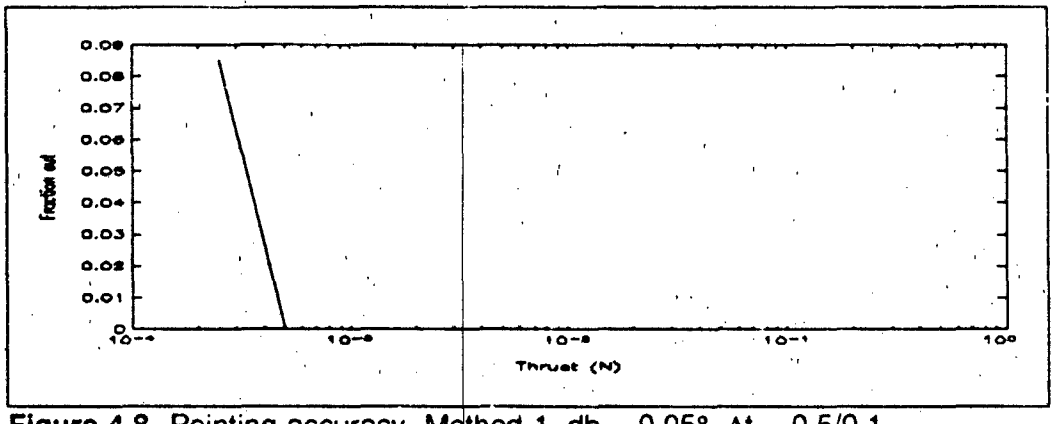

Figure 4.8 Pointing accuracy, Method 1, db = 0.05°, Δt = 0.5/0.1

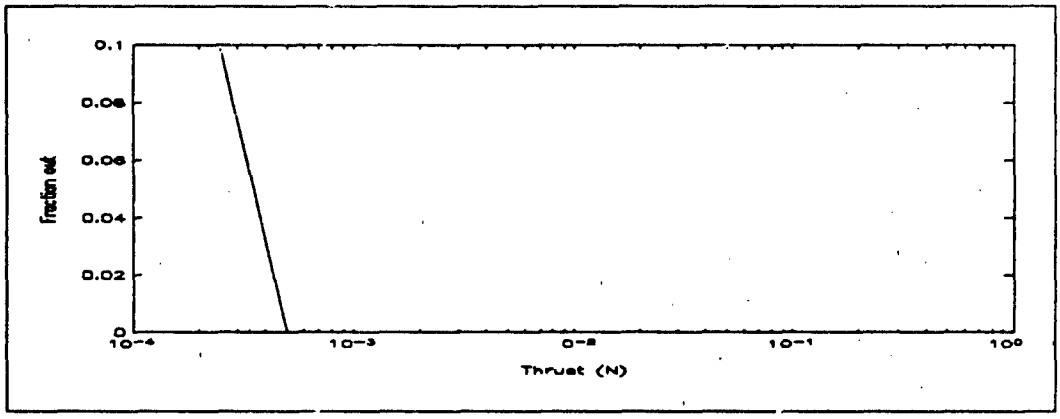

Figure 4.9 Pointing accuracy, Method 1, db = 0.02°, Δt = 0.5/0.1

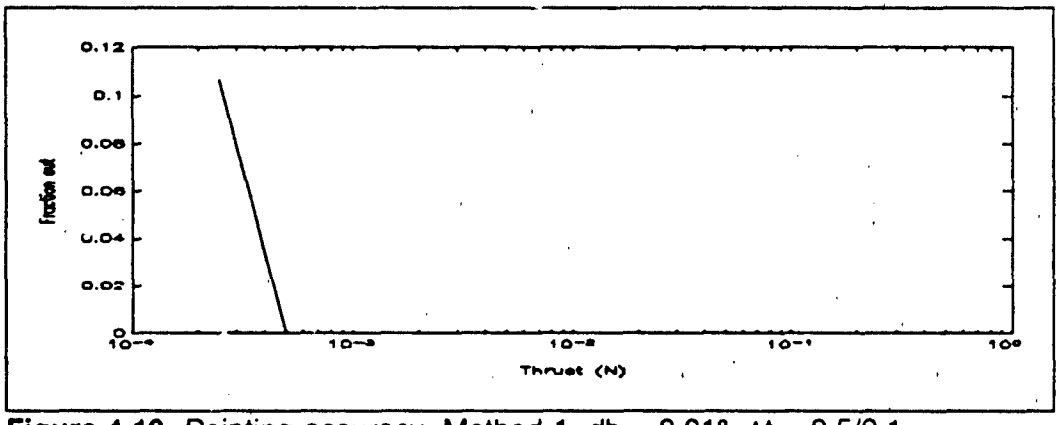

Figure 4.10 Pointing accuracy, Method 1, db = 0.01°, Δt = 0.5/0.1

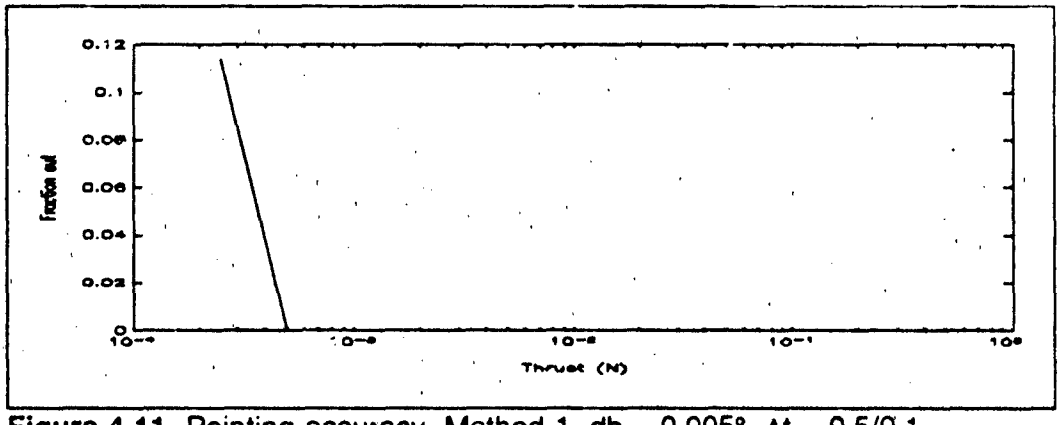

Figure 4.11 Pointing accuracy, Method 1, db = 0.005°, Δt = 0.5/0.1

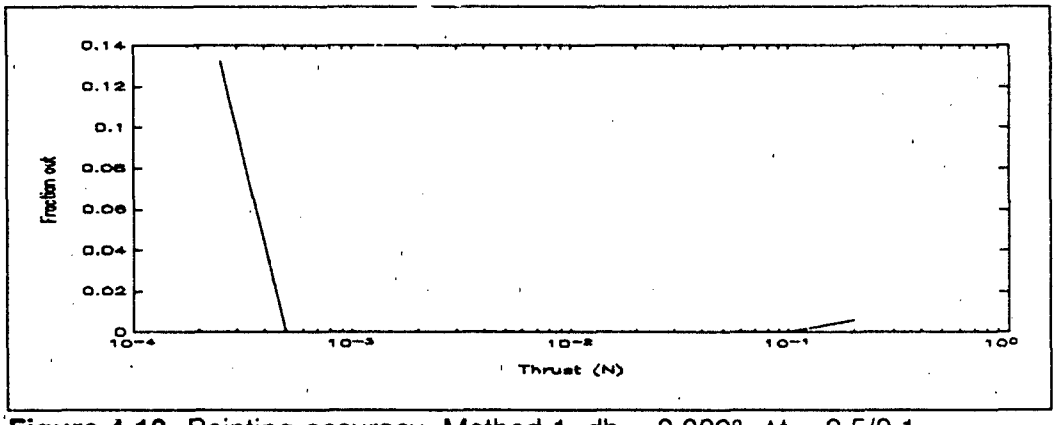

Figure 4.12 Pointing accuracy, Method 1, db = 0.002°, Δt = 0.5/0.1

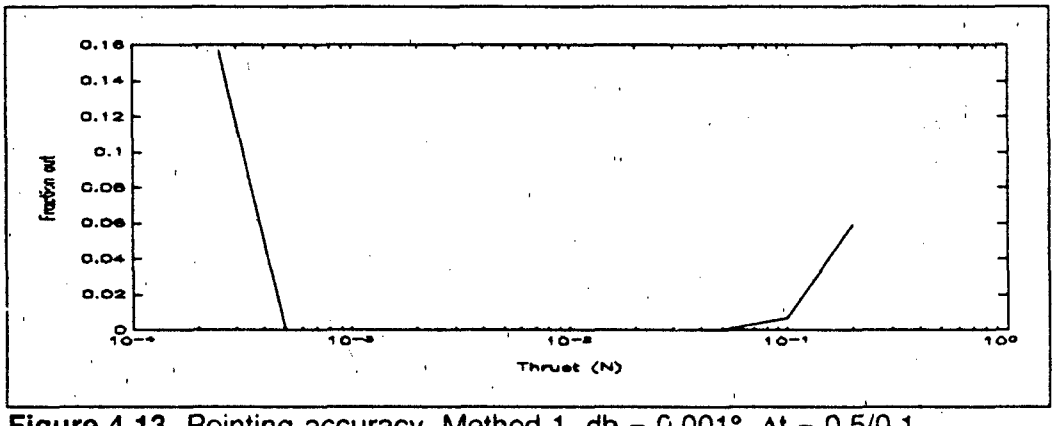

Figure 4.13 Pointing accuracy, Method 1, db = 0.001°, Δt = 0.5/0.1

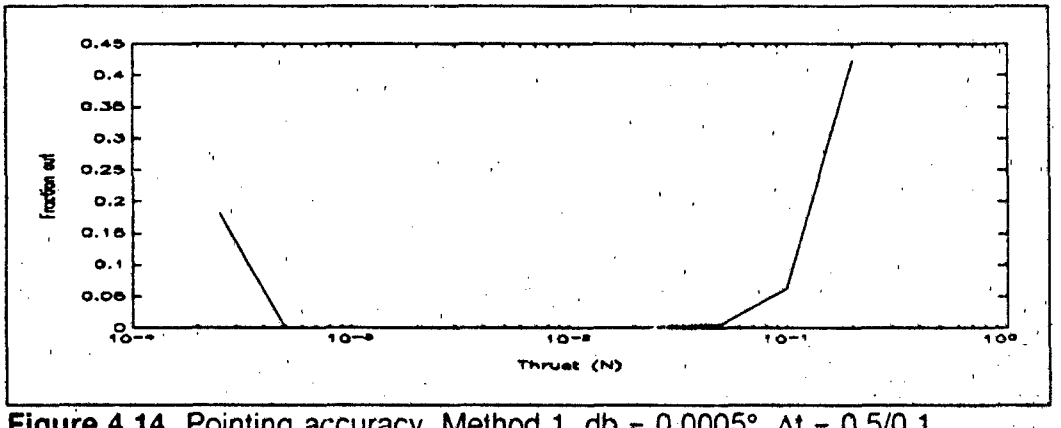

Figure 4.14 Pointing accuracy, Method 1, db = 0.0005°, Δt = 0.5/0.1

When using the momentum wheel plus one axis control, the error in ϕ follows the same pattern. The error in θ is dependent upon the controller, not the thrusters used. The error in ψ is out of the limits for most of the time. In order to maintain this axis within limits it is necessary to increase the angular momentum of the momentum wheel, thus providing greater gyroscopic stiffness.

It is clear from these graphs that accurate attitude control can be maintained with thrust levels as low as 0.5 mN per thruster. At the lowest thrust setting, 0.25 mN, the limits are exceeded approximately 14% of the time. This figure is independent of the pointing limits desired.

Thrust levels above this remain consistently within the limits, except for the at the narrowest limits, where the highest thrust settings lose the ability to maintain the satelliite attitude within limits. This is due to the fact that the minimum impulse bit is too large and the operating frequency too low for the limits specified. One pulse from a thruster results in an angular velocity that is too high to be consistently corrected before the error limit is exceeded. As a result, the spacecraft is unable to maintain pointing accuracy within the selected limits.

4.3 Total Angular Impulse Required

The total angular impulse required in order to maintain the spacecraft's attitude is also of interest, as it is this factor that determines the total mass of propellant required over the life of the satellite. In these graphs, the time that

each thruster pair was active is summed over the orbital period and multiplied times the moment delivered by the thruster to obtain a total angular impulse.

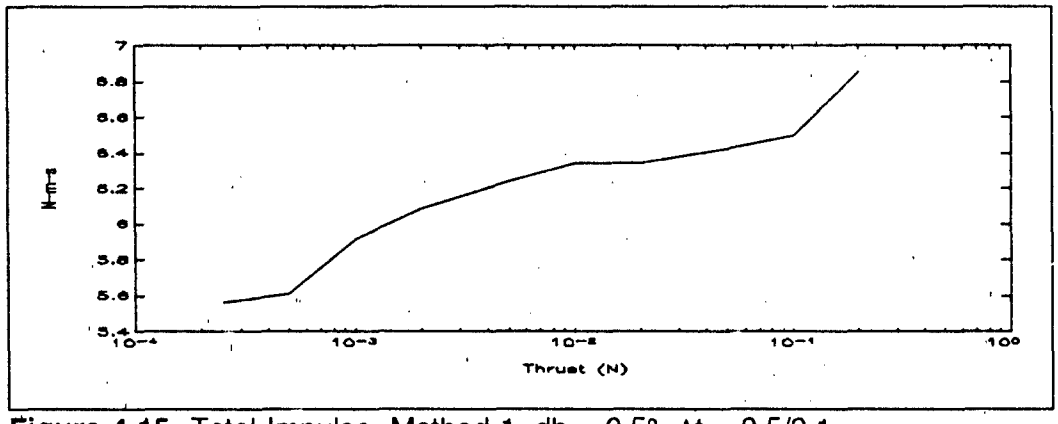

Figure 4.15 Total Impulse, Method 1, db = 0.5°, Δt = 0.5/0.1

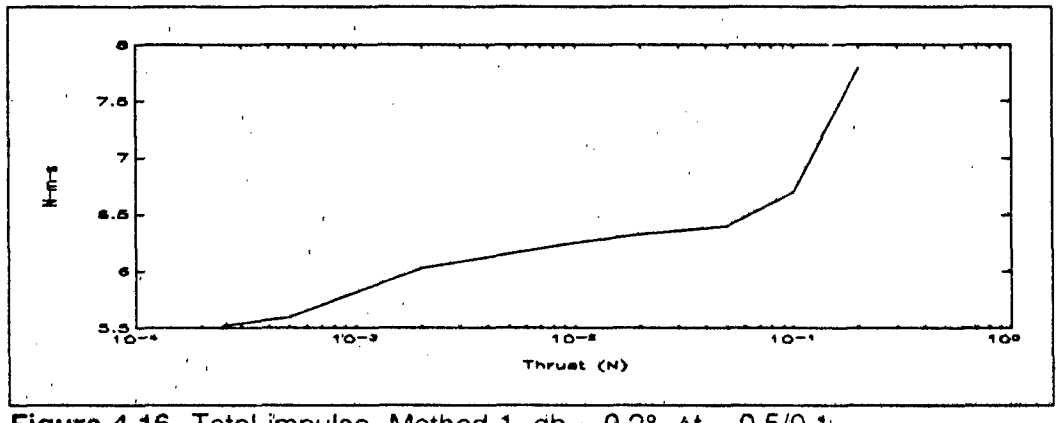

Figure 4.16 Total impulse, Method 1, db = 0.2°, Δt = 0.5/0.1

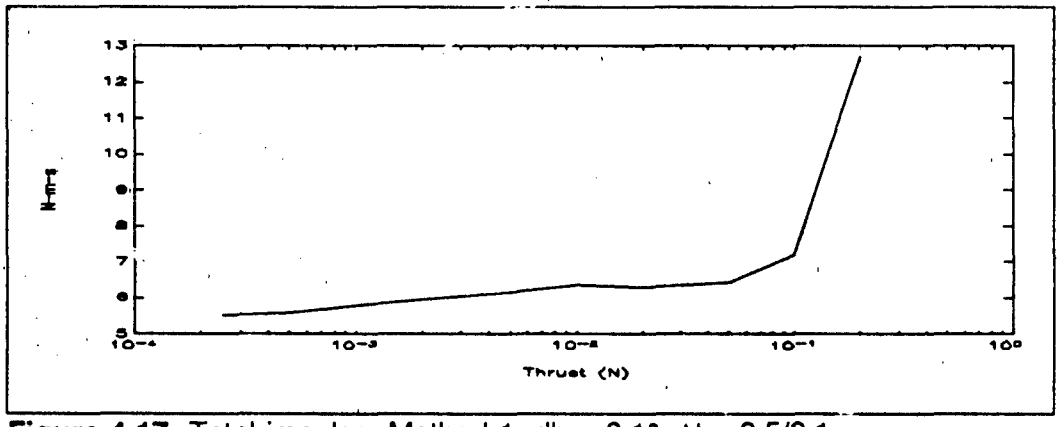

Figure 4.17 Total impulse, Method 1, db = 0.1°, Δt = 0.5/0.1

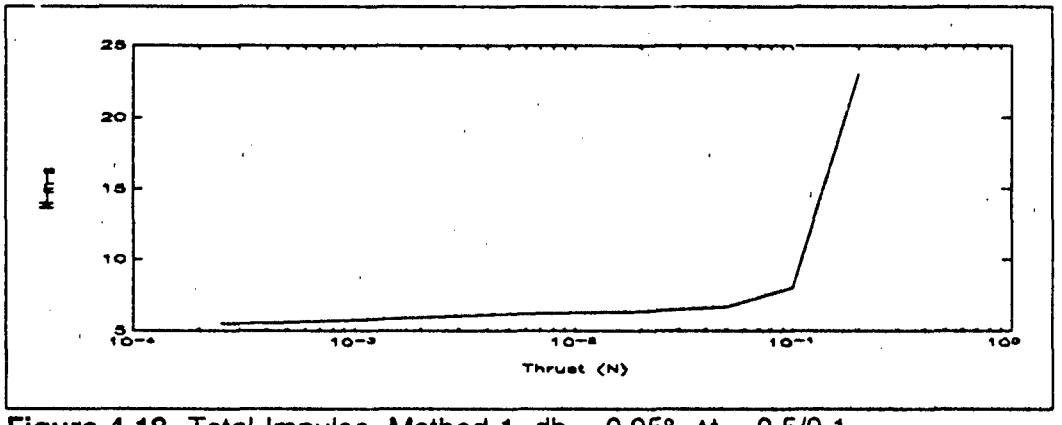

Figure 4.18 Total Impulse, Method 1, db = 0.05°, Δt = 0.5/0.1

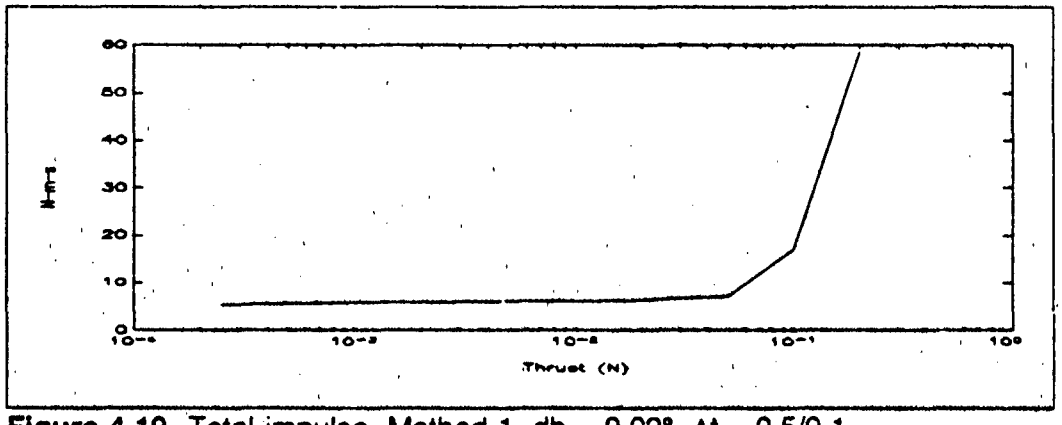

Figure 4.19 Total impulse, Method 1, db = 0.02°, Δt = 0.5/0.1

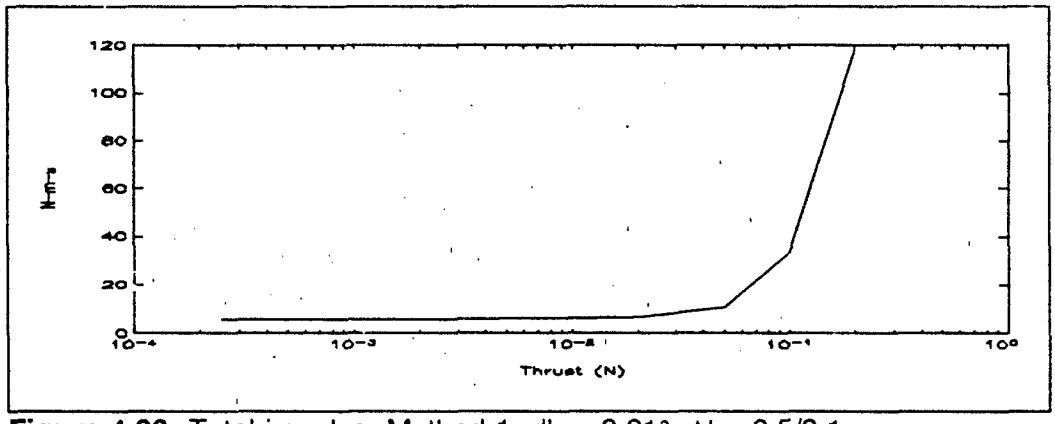

Figure 4.20 Total impulse, Method 1, db = 0.01°, Δt = 0.5/0.1

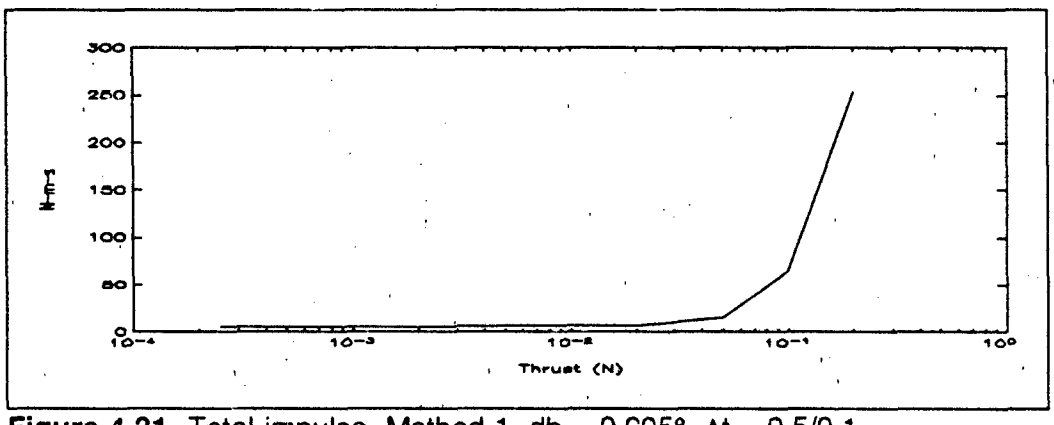

Figure 4.21 Total impulse, Method 1, db = 0.005°, Δt = 0.5/0.1

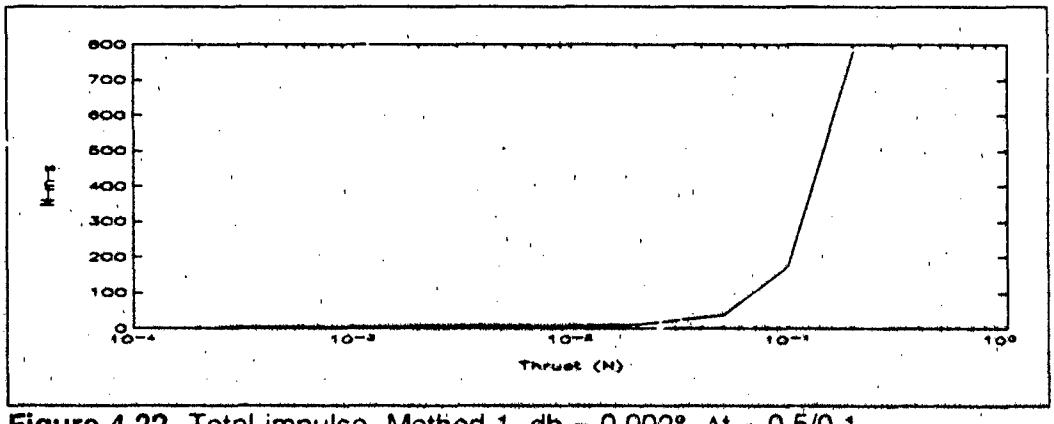

Figure 4.22 Total impulse, Method 1, db = 0.002°, Δt = 0.5/0.1

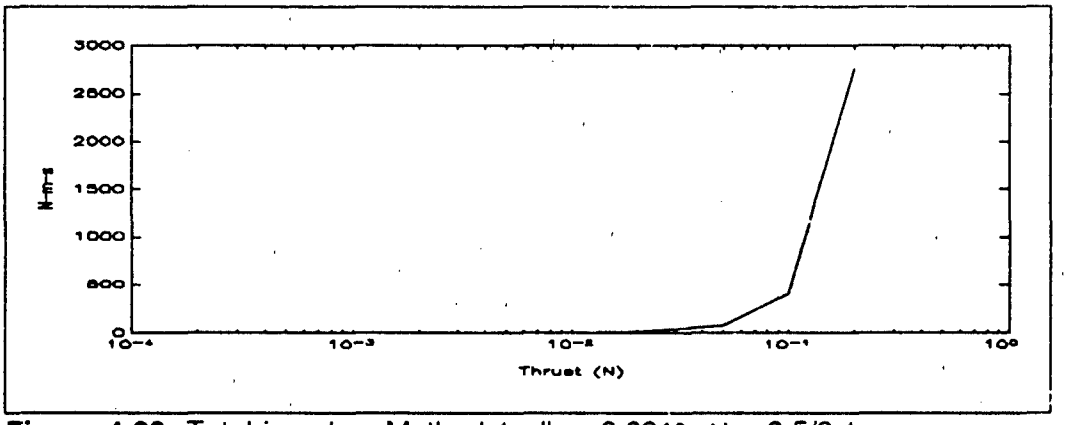

Figure 4.23 Total impulse, Method 1, db = 0.001°, Δt = 0.5/0.1

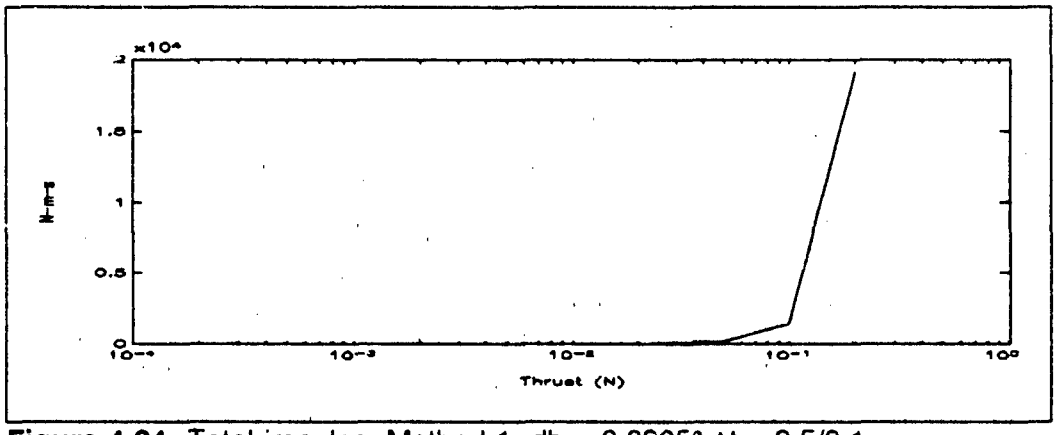

Figure 4.24 Total impulse, Method 1, db = 0.0005°,Δt = 0.5/0.1

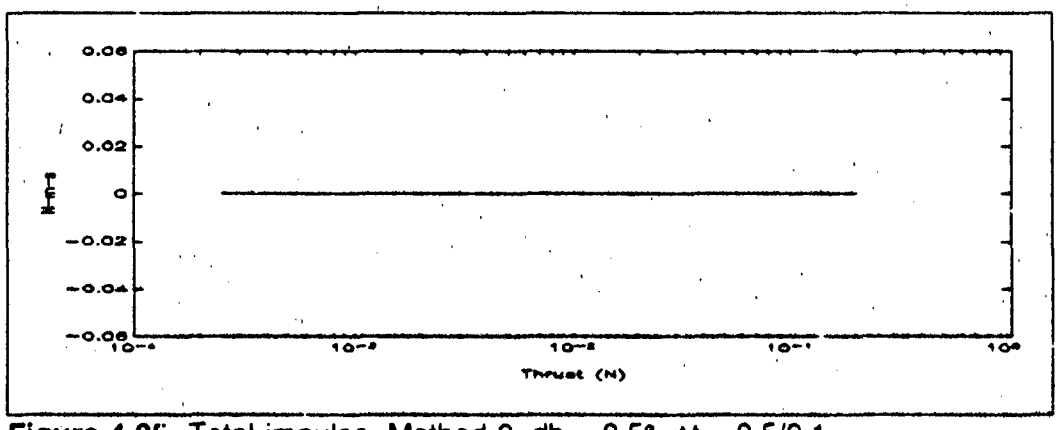

Figure 4.25 Total impulse, Method 2, db = 0.5°, Δt = 0.5/0.1

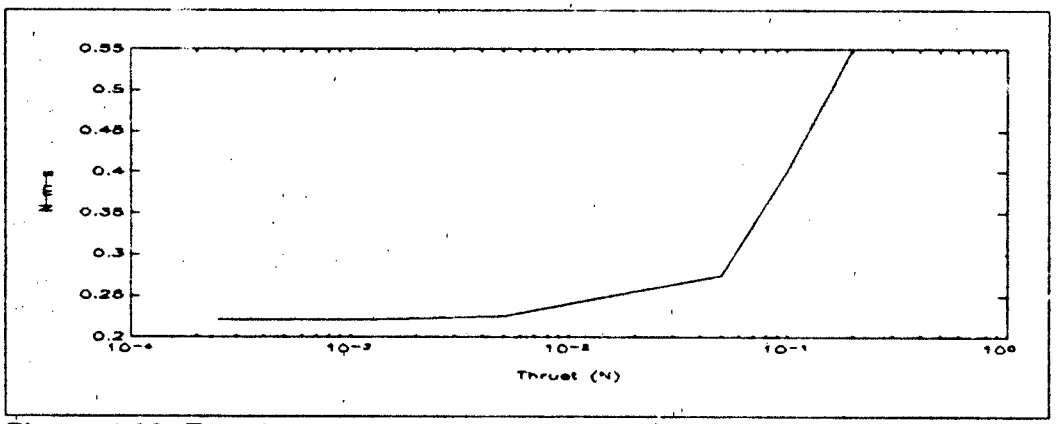

Figure 4.26 Total impulse, Method 2, db = 0.2°, Δt = 0.5/0.1

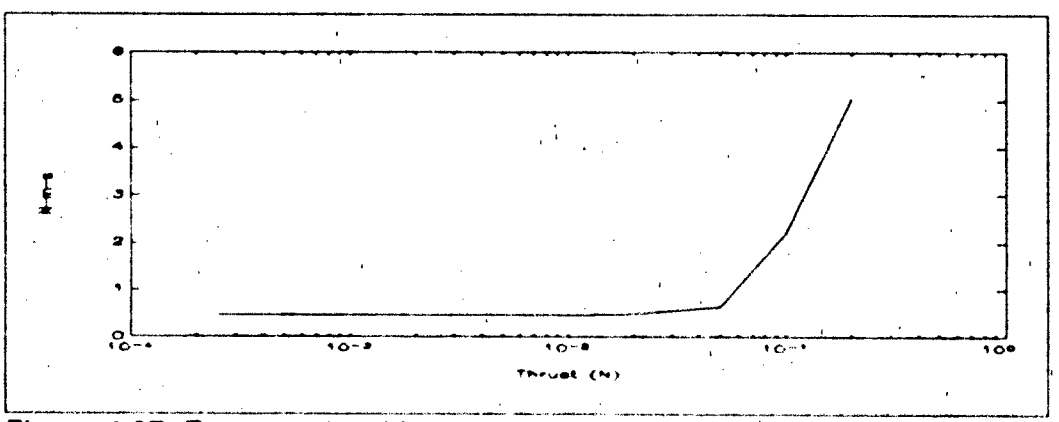

Figure 4.27 Total impulse, Method 2, db = 0.1°, Δt = 0.5/0.1

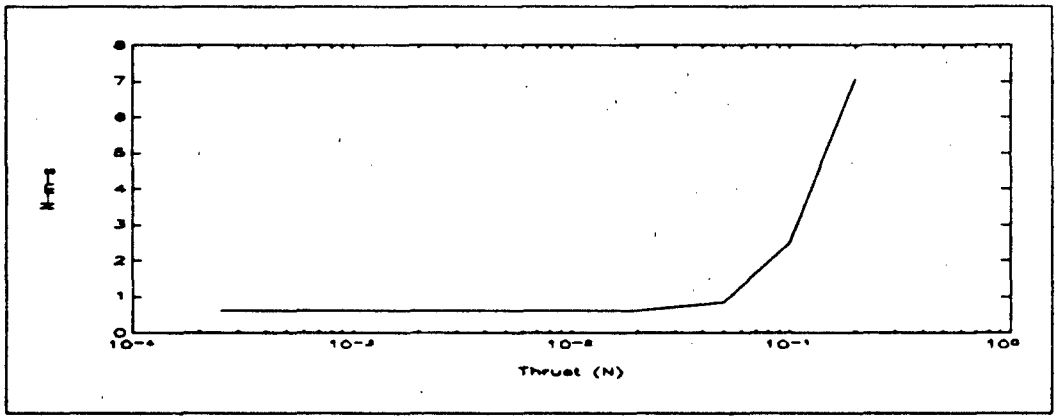

Figure 4.28 Total impulse, Method 2, db = 0.05°, Δt = 0.5/0.1

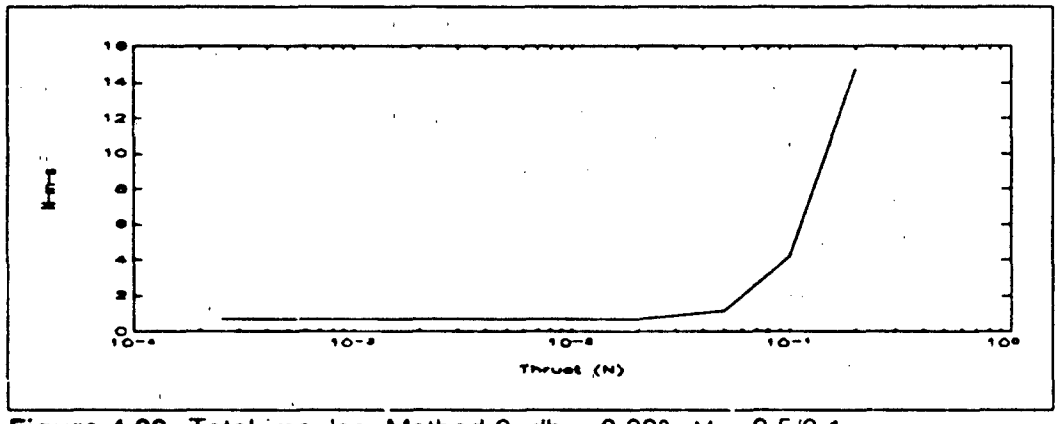

Figure 4.29 Total impulse, Method 2, db = 0.02°, Δt = 0.5/0.1

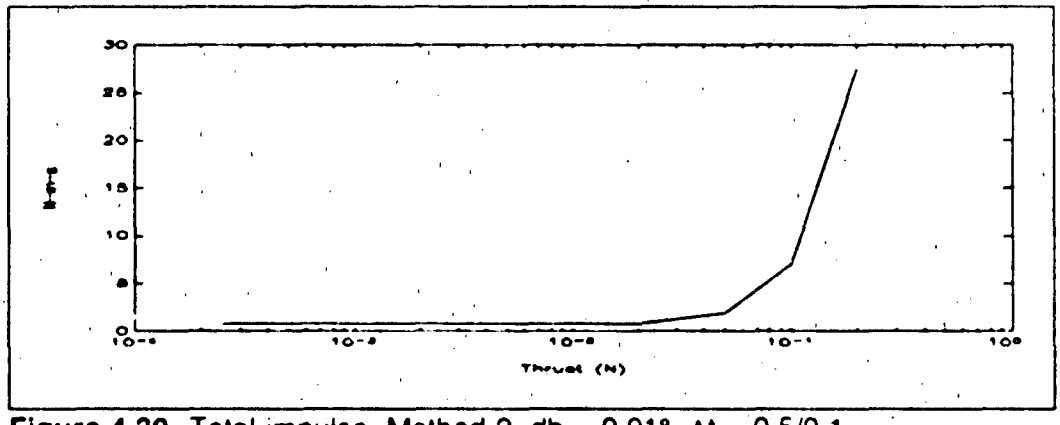

Figure 4.30 Total impulse, Method 2, db = 0.01°, Δt = 0.5/0.1

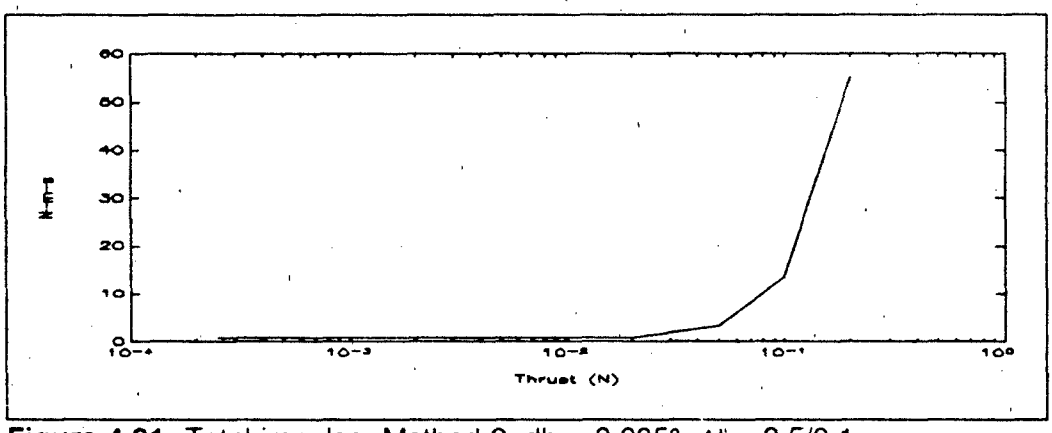

Figure 4.31 Total impulse, Method 2, db = 0.005°, Δt = 0.5/0.1

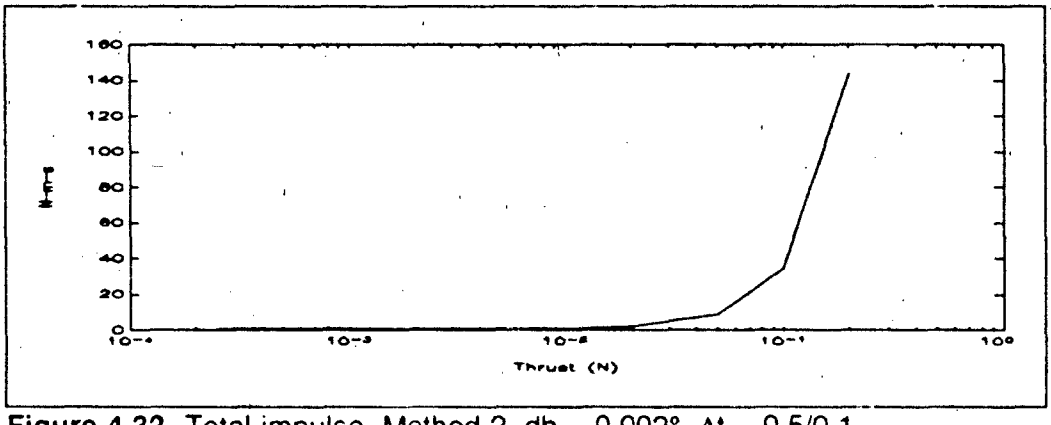

Figure 4.32 Total impulse, Method 2, db = 0.002°, Δt = 0.5/0.1

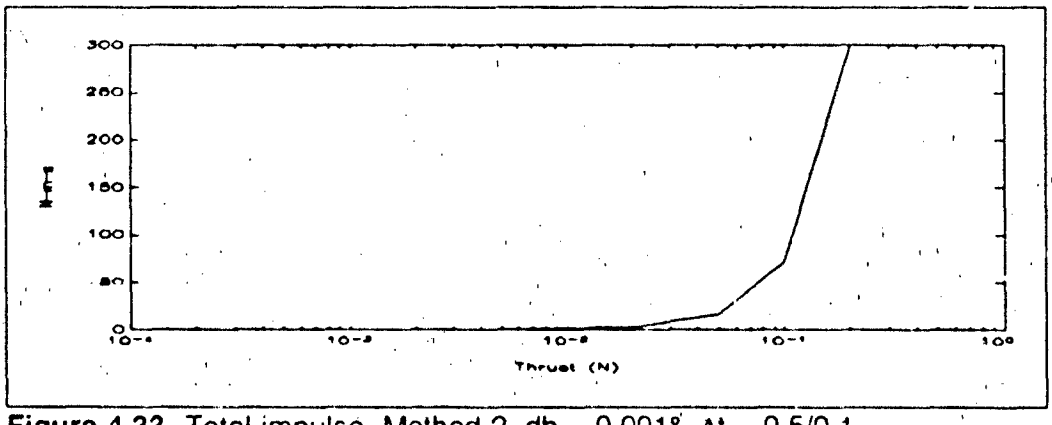

Figure 4.33 Total impulse, Method 2, db = 0.001°, Δt = 0.5/0.1

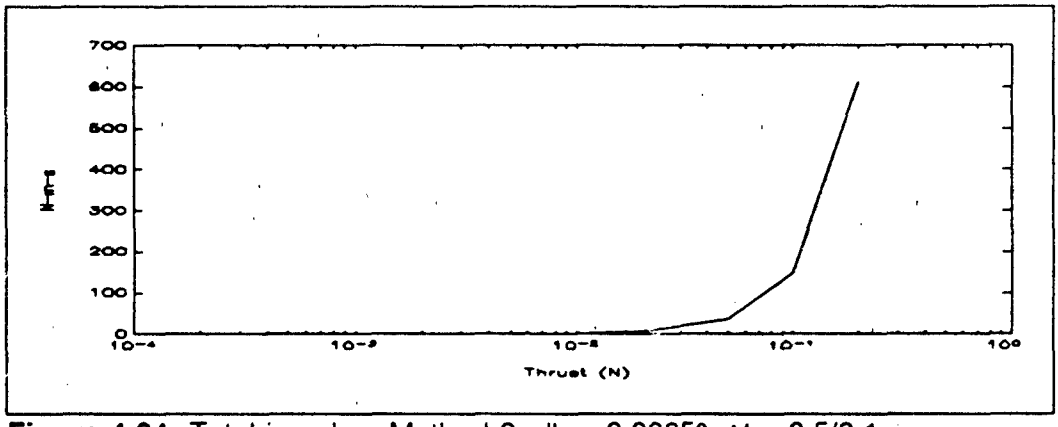

Figure 4.34 Total impulse, Method 2, db = 0.0005°, Δt = 0.5/0.1

The total angular impulse can be seen to be near constant for a given deadband as long as the thrust level is low. At higher thrust levels, the total angular impulse required increases sharply, because of the larger minimum impulse bit. Where the total is constant, the thrusters are merely acting to push the satellite axis away from one side of the deadband, and allowing the perturbative moments to force it back. As the thrust level is increased, a single pulse from the thruster drives the axis to the other side of the deadband, causing the opposing thruster to fire and drive it back. Instead of the thrusters being used to maintain only one side of the deadband, they are used on both sides, resulting in a reduction in efficiency. At still higher levels of thrust, the axis traverses the deadband in a shorter time, requiring the thrusters to fire more often. The limiting case is reached when the thrusters are firing during every time step. When this occurs, the system loses the ability to maintain the attitude within limits.

Figure (4.25) is of special interest, as it represents a set of conditions that requires no thruster firings during the period observed. Due to the gyroscopic effect of the momentum wheel, the small magnitudes of the perturbing moments cannot cause the attitude error in ϕ to exceed the limits within one orbital period.

4.4 Effects of Smaller Step Size

When examining the data already collected, it becomes obvious that the limiting factor is the minimum impulse bit. By reducing the step size, the minimum duration of one thruster firing can be reduced, thus reducing the minimum impulse bit. The next series of graphs show the data changes caused in the system effectiveness by reducing the step size. All data covers a period of one-sixth of an orbit. The first set, figures (4.35-41) uses a deadband of 0.0005°, while varying the step size. The pointing accuracy results for a step size of 0.05/0,01 seconds is not included as it is identical to the results with a step size of 0.1/0.02 seconds. In both of these cases, the system is able to maintain itself within the limits with 100% accuracy.

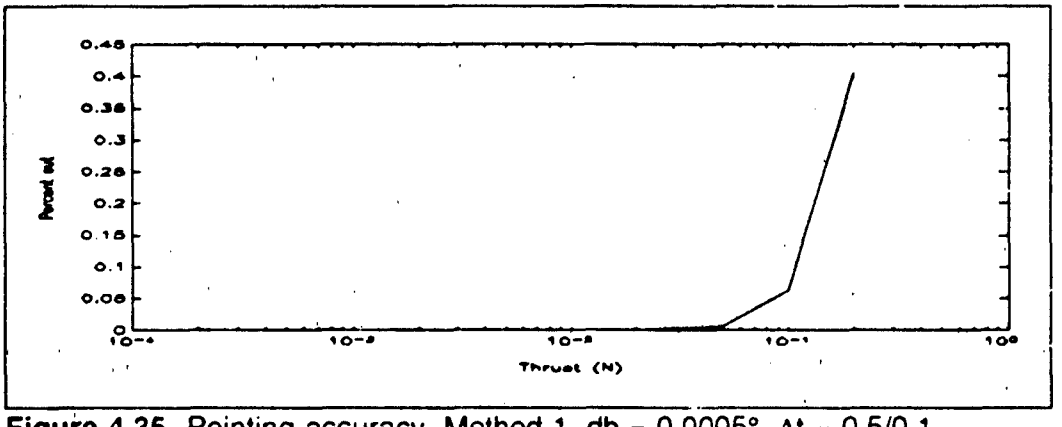

Figure 4.35 Pointing accuracy, Method 1, db = 0.0005°, Δt = 0.5/0.1

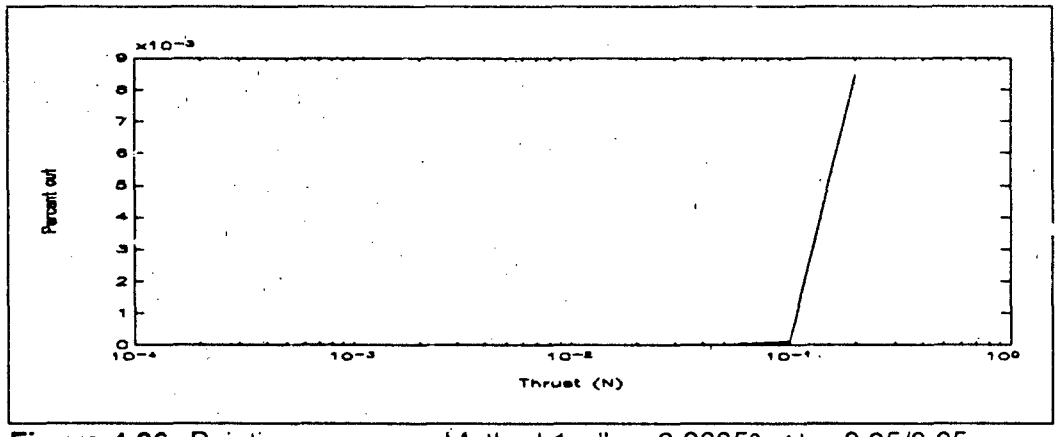

Figure 4.36 Pointing accuracy, Method 1, db = 0.0005°, Δt = 0.25/0.05

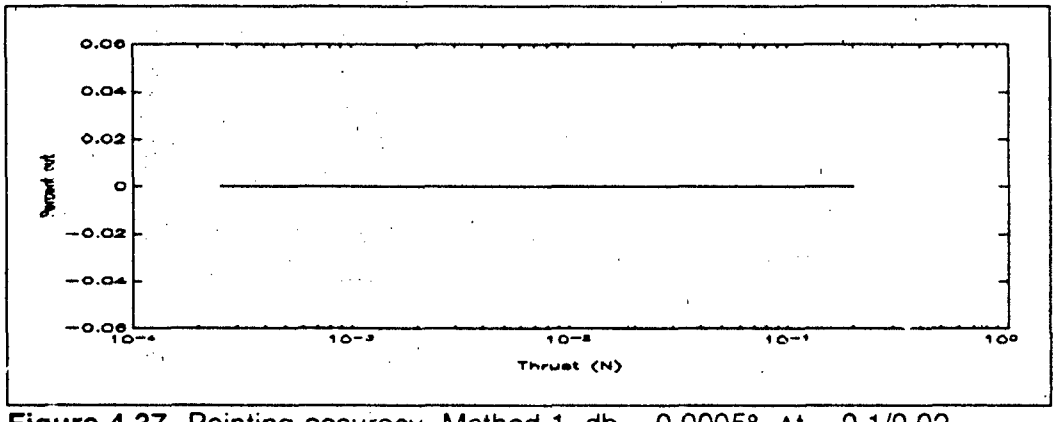

Figure 4.37 Pointing accuracy, Method 1, db = 0.0005°, Δt = 0.1/0.02

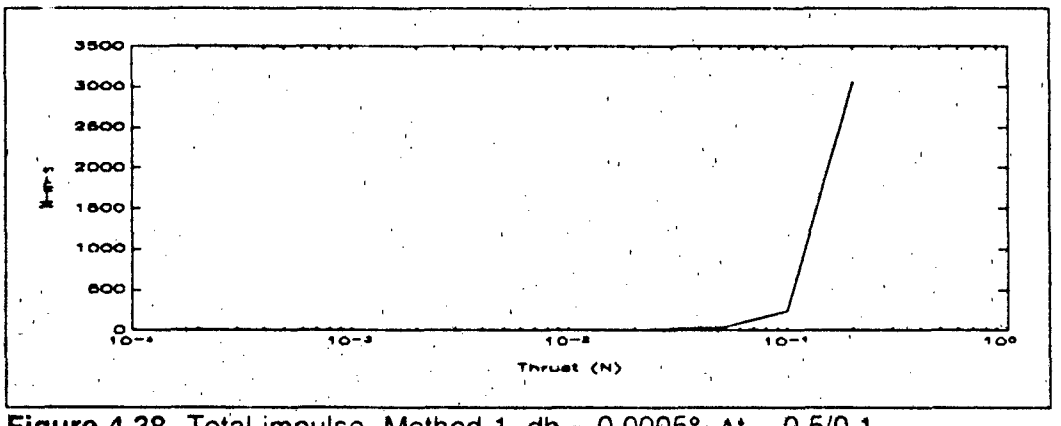

Figure 4.38 Total impulse, Method 1, db = 0.0005°, Δt = 0.5/0.1

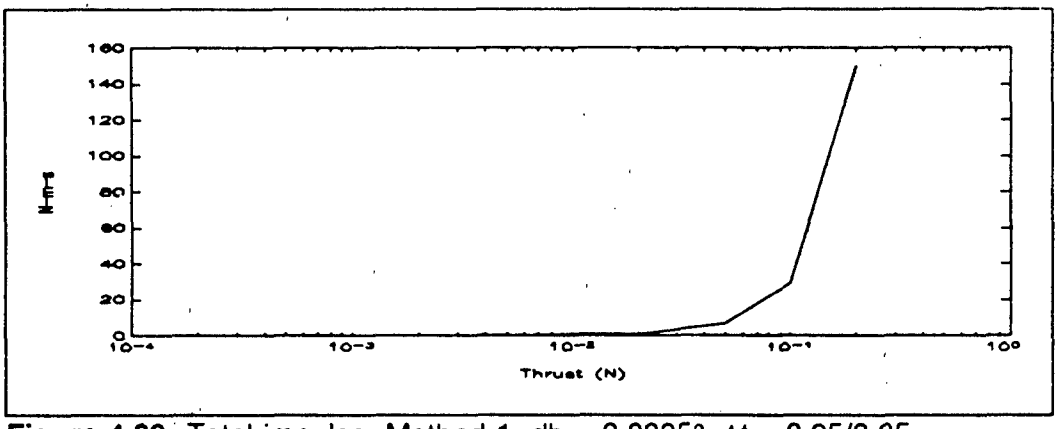

Figure 4.39 Total impulse, Method 1, db = 0.0005°, Δt = 0.25/0.05

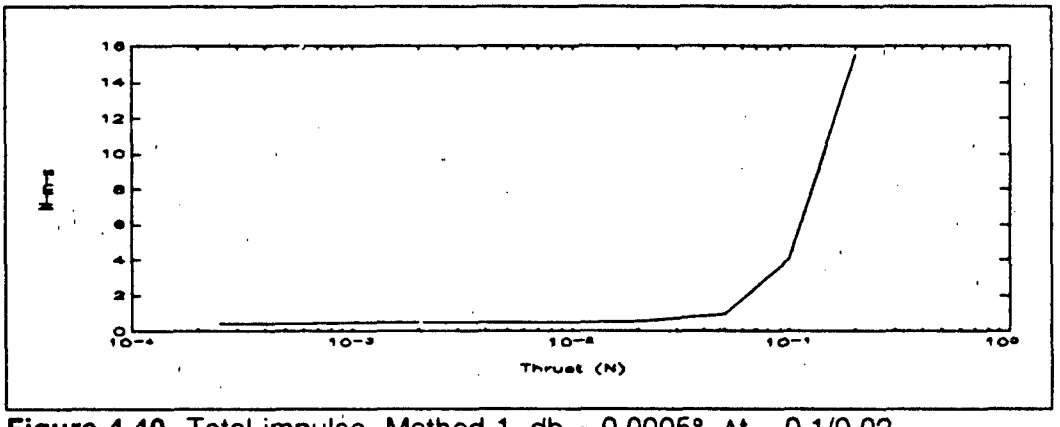

Figure 4.40 Total impulse, Method 1, db = 0.0005°, Δt = 0.1/0.02

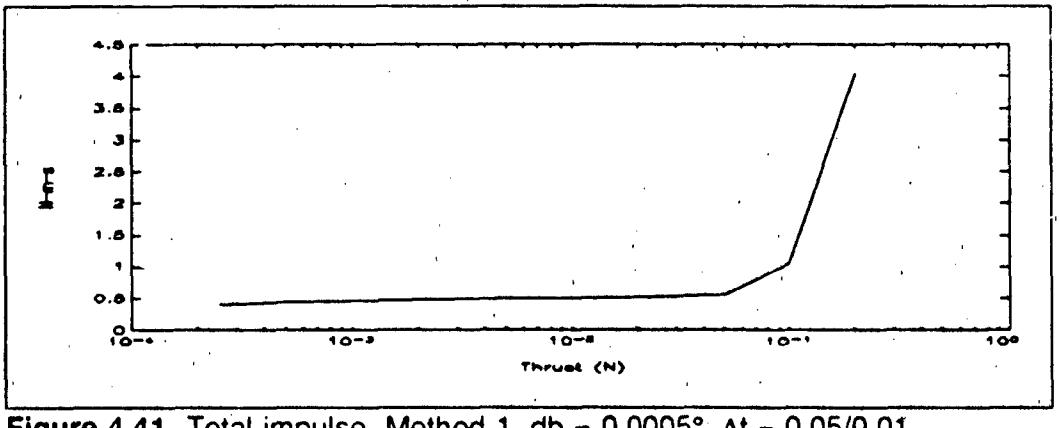

Figure 4.41 Total impulse, Method 1, db = 0.0005°, Δt = 0.05/0.01

As a final look at the effect of step size on system effectiveness, data is presented on performance with error limits of 0.000002°, or approximately one-hundredth of an arcsecond. The data covers a period of one-sixth of an orbit, and is presented in figures (4.42-49).

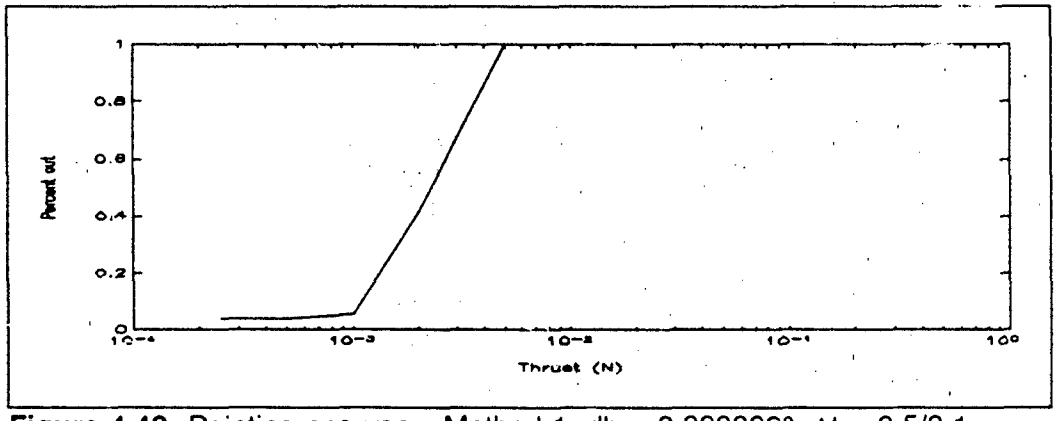

Figure 4.42 Pointing accuracy, Method 1, db = 0.000002°, Δt = 0.5/0.1

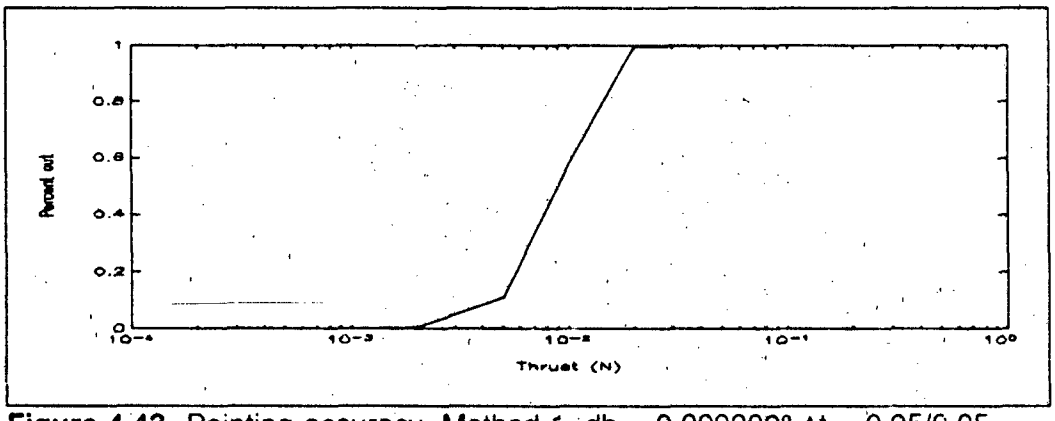

Figure 4.43 Pointing accuracy, Method 1, db = 0.000002°,Δt = 0.25/0.05

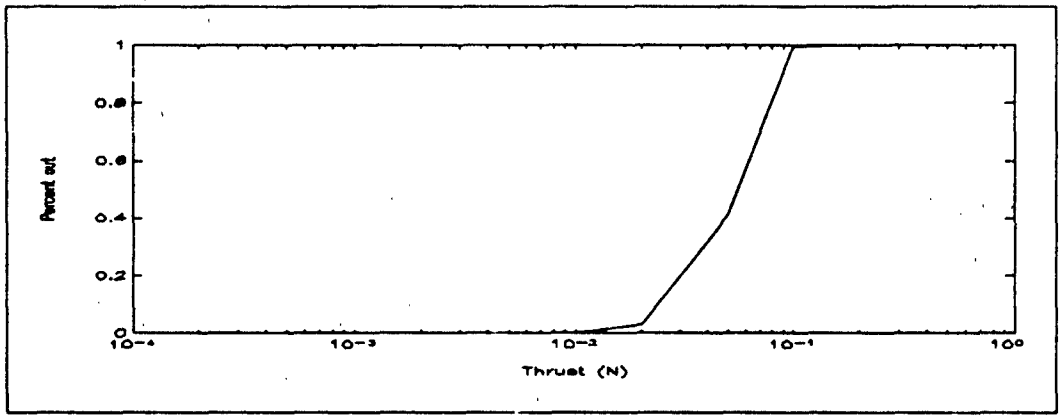

Figure 4.44 Pointing accuracy, Method 1, db = 0.000002°, Δt = 0.1/0.02

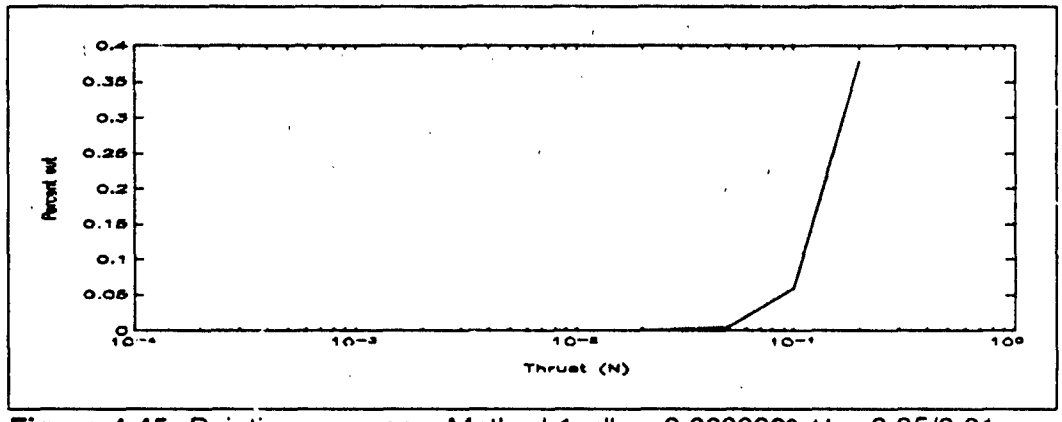

Figure 4.45 Pointing accuracy, Method 1, db = 0.000002°,Δt = 0.05/0.01

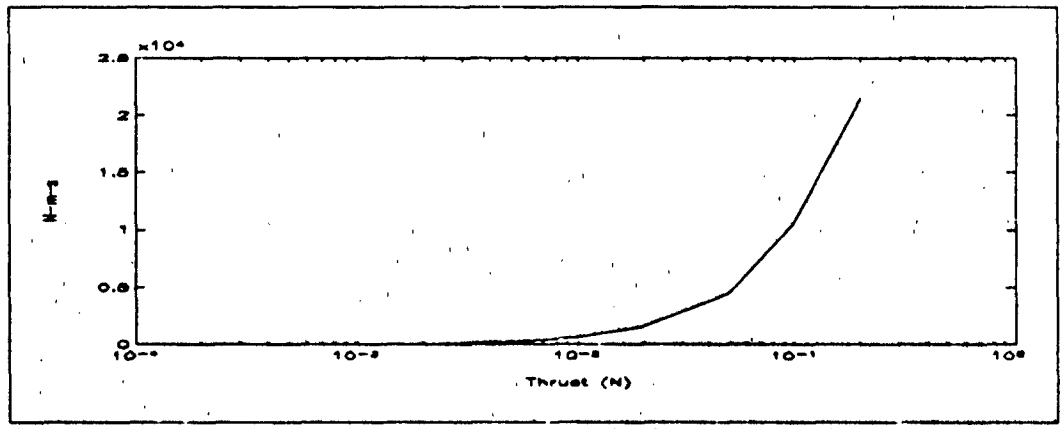

Figure 4.46 Total impulse, Method 1, db = 0.000002°, Δt = 0.5/0.1

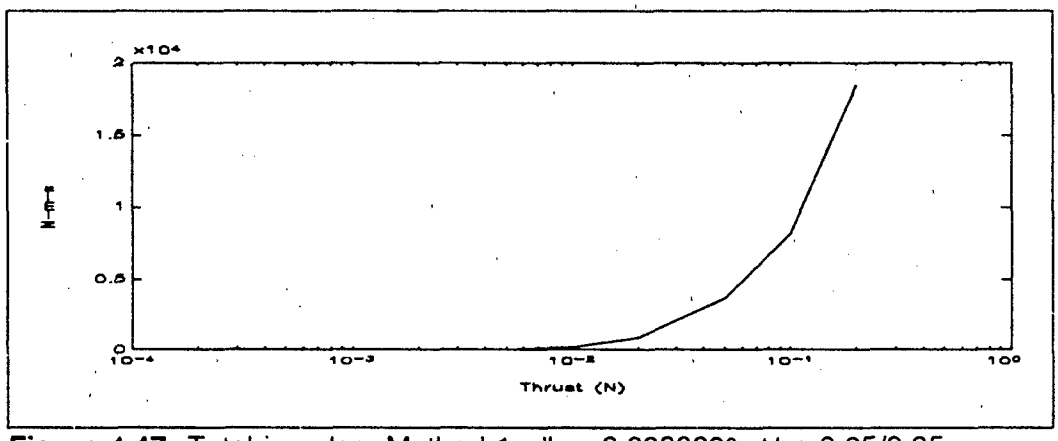

Figure 4.47 Total impulse, Method 1, db = 0.000002°, Δt = 0.25/0.05

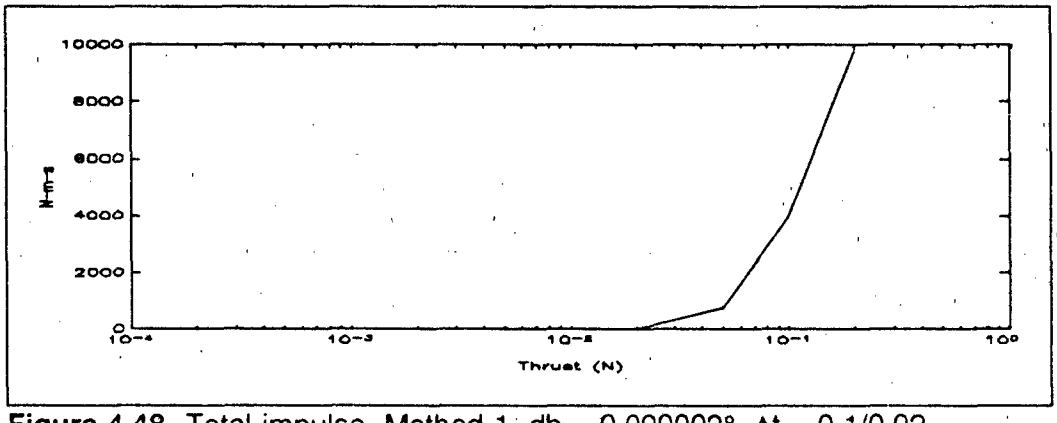

Figure 4.48 Total impulse, Method 1, db = 0.000002°, Δt = 0.1/0.02

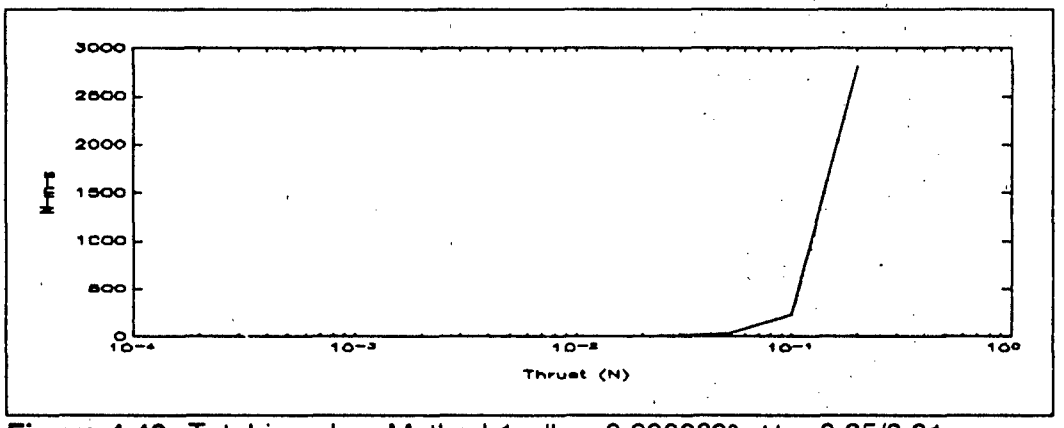

Figure 4.49 Total impulse, Method 1, db = 0.000002°, Δt = 0.05/0.01

From the graphs, it can be seen that none of the levels of thrust examined offer acceptable performance if a time step of 0.5/0.1 is used, as all result in the attitude being out of limits during over 98% of the control system sampling times. At shorter time steps, it becomes possible to achieve adequate performance, and each further reduction in the time step results in higher levels of thrust being acceptable.

As a result of the data collected, it can be seen that the two factors of greatest importance in the performance of the attitude control are the minimum angular impulse delivered and the frequency of the controller. Either reducing the change in angular momentum per pulse or increasing the frequency of control actions allows narrower limits to be met. The level of thrust provided has no effect on the ability of the system, except in as much as it affects the minimum impulse bit and as long as it surpasses the perturbative forces acting on the spacecraft. If the thrust level is decreased beyond that required to

achieve an acceptable minimum angular impulse, there is no effect on the total angular impulse required to maintain proper attitude control. Thus it has no effect upon the mass of propellant required, since the propellant required to deliver a given total angular impulse is a function of the thruster specific impulse and position, not of its thrust. This is true in all of the combinations of factors examined in this analysis.

4.5 Response Time

The analysis of the response times is brief. The results of the data collected indicates that the nonlinearities, axis coupling, and perturbations have very little effect upon the time required for the system to recover from an attitude error about one axis. Instead, all responses can be approximated by Newtonian dynamics, where the angular acceleration equals the applied moments divided by the Moment of Inertia of that axis. Thus, doubling the thrust will reduce the time required to correct an error by the square root of two. Verification of this treatment is provided by the following plots of response times shown in figures (4.50-52). Each graph shows three curves: one with the perturbative forces opposing the corrective action, one with the perturbative forces assisting the corrective action, and the third generated from a Newtonian treatment of the control laws. In each graph, the Newtonian approximation results in a value between the other two.

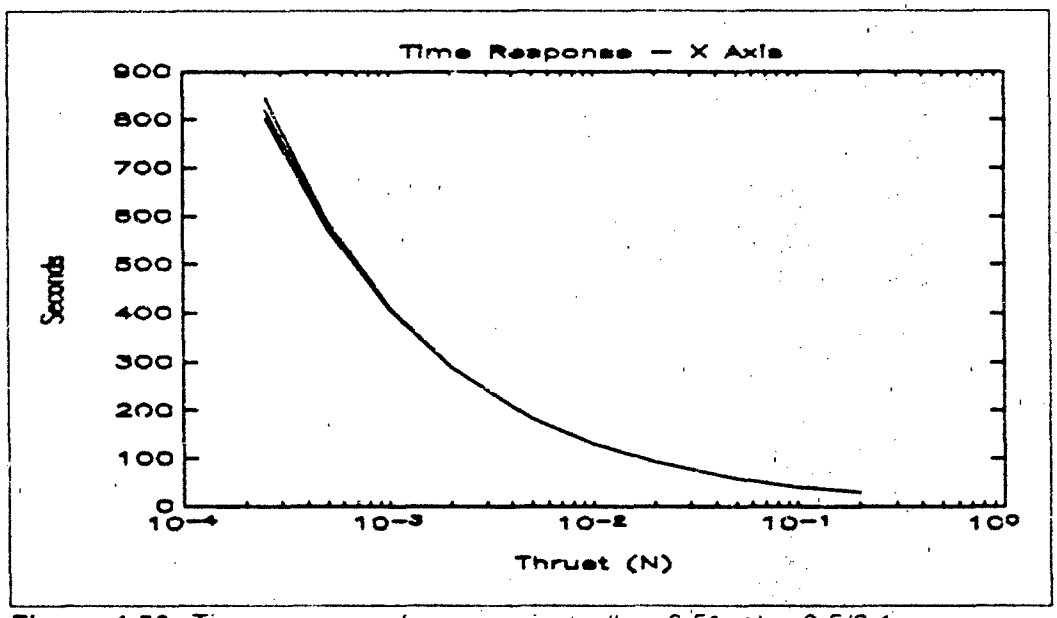

Figure 4.50 Time response for errors in ϕ, db = 0.5°, Δt = 0.5/0.1

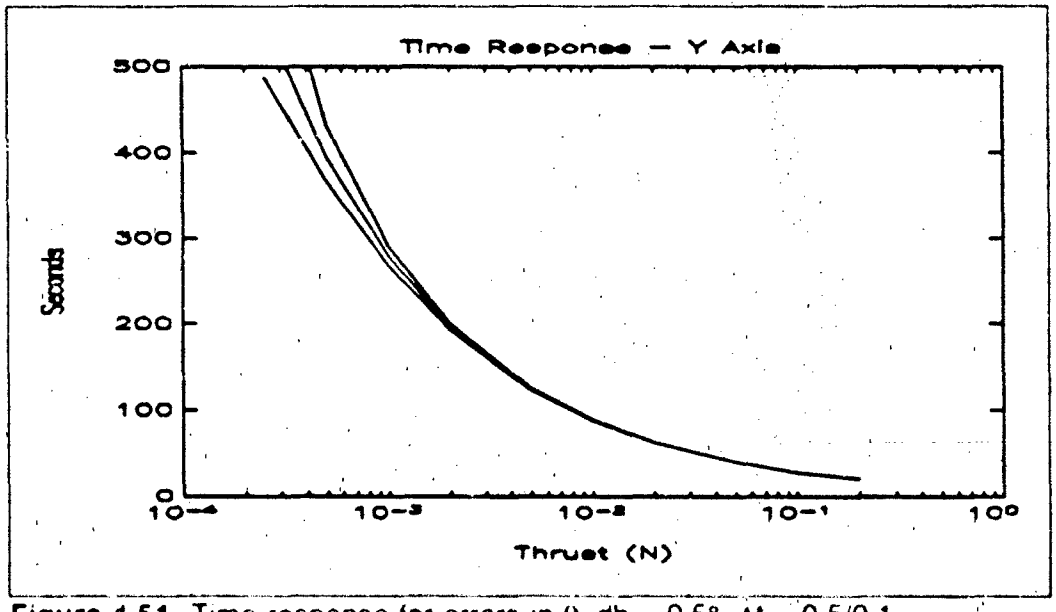

Figure 4.51 Time response for errors in θ, db = 0.5°, Δt = 0.5/0.1

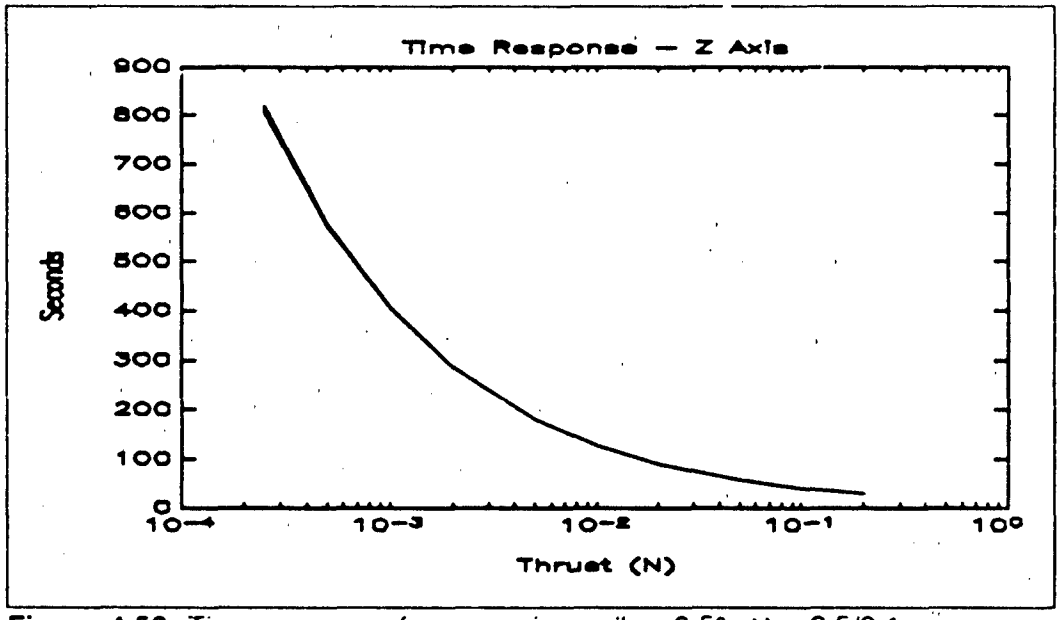

Figure 4.52 Time response for errors in ψ, db = 0.5°, Δt = 0.5/0.1

V. Conclusion

In order to investigate the use of low thrust propulsion systems using numerical integration techniques, it is necessary to use a very small step size. The impulse bit of the thruster as applied by this analysis is limited to no less than the thrust times the step size. Therefore, in order to examine the advantages of low thrust systems versus chemical rockets, the step size must approximate the thruster pulse durations. At the same time, the low angular accelerations experienced means that relatively long periods of time must be covered in the integration limits in order to examine cyclic behavior. These two facets combine to require a large amount of computer processing time. If the effects of variations in the perturbative moments are to be considered, the duration covered by the integration must be still longer, since these effects follow a cycle with a period of one day.

However, such integrations are not necessary in the design process. The primary figures of merit in a propulsion system for attitude control are the minimum impulse bit and the pulse rate. Beyond that, as long as the thrusters produce greater moments than those caused by the perturbations, the thrust of the system has no effect upon its efficiency or accuracy. Thus low thrust systems can be treated the same as chemical propulsion systems of much higher thrust. The propellant required in either case is determined by the total impulse required divided by the specific impulse of the thruster. In this case, a low thrust system might have a mass advantage due to its higher specific

impulse and correspondingly lower mass of propellant required. However, the decision must be based upon total mass of the system, including the thruster itself and any supporting equipment. The equations for determining the optimum specific impulse for a given total impulse are well understood and in common use in satellite design.

The influence of response times upon the system design is dependent upon the purpose and requirements of the spacecraft. A communications satellite, such as the Intelsat VII has no need to perform rapid attitude changes during normal operations, and a low thrust system is adequate for its use. However, during initial deployment of the spacecraft, some supplementary method may be needed for in order to allow the spacecraft to attain an Earth pointing attitude within a reasonable time frame. At the lowest thrust level examined in this analysis, a satellite the size of Intelsat VII would require over one hundred and seventy hours to halt a rotation of one revolution per minute. If the satellite cannot deploy its solar arrays until the attitude has stabilized, all of this maneuvering must be performed using battery power. Therefore the restrictions on how long the deployment and attitude acquisition sequence can take may restrict the minimum thrust allowed, require additional despin thrusters to halt spacecraft spin, or require that spacecraft injection into the geosynchronous orbit be accomplished with near zero residual angular motion.

The results of this analysis indicate that low thrust systems are capable of accurate attitude control. Their treatment in designing an attitude control

system should be no different than that of chemical propulsion. In both cases, the limiting factor is the minimum impulse bit of the thruster and the operations rate of the system. Further examination of this subject is not recommended.

```
      PROGRAM thesis
c     This program uses a Hamming Integrator to apply perturbations to
c     satellite dynamics.  Thruster firings are continuous, with haming
c     being reinitialized as thrusters turn on and off.
      implicit double precision(a-h,m,o-z)
      double precision Ix,Iy,Iz,nu,thrusts(10),nun
      data thrusts/0.2,0.1,0.05,0.02,0.01,0.005,0.002,0.001,0.0005,
     1    0.00025/
      logical flagxp,flagxm,flagyp,flagym,flagzp,flagzm
c
      common /ham/ t,x(12,4),f(12,4),errest(12),n,h
      common /wheel/ hx,hy,hz,hxdot,hydot,hzdot
      common /const/ Ix,Iy,Iz,w0,w0dot,ecc
      common /state/ phi,theta,psi,phidot,thetadot,psidot
      common /moments/ db,thrust,mcx,mcy,mcz,mx,my,mz
      common /sun/ alpha0,delta,Sx,Sy,Sz
      common /nominal/ phi0,theta0,psi0
      common /dbug/ xblock,yblock,zblock,wx,wy,wz
c
c     OPEN OUTPUT FILES
c
      open(15,file='th2a.o',status='unknown')
      open(14,file='lim2a.o',status='unknown')
c
c     SET CONSTANTS
c
      Ix = 8000
      Iy = 3700
      Iz = 7850
      alpha0 = 0.0
      psi0 = 0.0
      theta0 = 0.0
      phi0 = 0.0
      delta =23.443597*3.1415962/180.0
      ecc = 0.01
      db = 0.5*3.1415962/180.0
      h = 0.5

c     WRITE HEADERS TO OUTPUT FILES
      write (15,43) db
      write (14,43) db
 43   format(2x,' Deadband= ',f10.8,', Step= 0.5, 0.1')
c
```

```
c     SET UP TIMING AND DIMENSIONS FOR HAMING
c
      n = 10
      t = 0.D 00
      tf = ((23*60+56))*60
      nstep = int((tf-t)*h)
c
c     INTEGRATE OVER ONE PERIOD FOR EACH LEVEL OF THRUST
c
      do 300 k1 = 1,10
c
c      *** INITIAL CONDITIONS ***
c
       thrust = thrusts(k1)
       arm = 1.25
       gain = 2*thrust*arm/db
       tau = 2*sqrt(Ix/gain)
       meanmot = 2*3.1415962/((23*60+56)*60)
       zeta = datan(2*sqrt(Iz*meanmot/35))
       x(1,1)=0
       x(2,1)=0
       x(3,1)=0
       x(4,1)=0
       x(5,1)=0
       x(6,1)=0
       x(7,1)=0
       x(8,1)=-35
       x(9,1)=0
       x(10,1)=0
       hy = x(8,1)

c      ZERO OUT PERFORMANCE STATISTICS
       onx=0
       ony=0
       onz=0
       offx=0
       offy = 0
       offz = 0
       inx = 0
       iny = 0
       inz = 0
       outx = 0
       outy = 0
       outz = 0
```

```
      outtot = 0
      flagxp = .false.
      flagxm = .false.
      flagyp = .false.
      flagym = .false.
      flagzp = .false.
      flagzm = .false.
      mcx = 0
      Mcy = 0
      Mcz = 0
      ct = 0
c
c     *** INITIALIZE HAMING ***
c
      nxt = 0
      call haming(nxt)
      if(nxt .ne. 0) go to 50
       write (15,1)
 1      format(2x,' haming did not initialize')
       stop
  50   continue
c
c     *** INTEGRATION LOOP ***
c
 200  continue
     if (ct .ge. tf) then
       goto 100
      else

     phi =   x(1,nxt)
     theta = x(2,nxt)
     psi =   x(3,nxt)
     phidot =   x(4,nxt)
     thetadot = x(5,nxt)
     psidot =   x(6,nxt)
     hx = x(7,nxt)
     hy = x(8,nxt)
     hz = x(9,nxt)

c    CONTROL SYSTEM OPERATIONS

c    SET TRIGGERS
     tr1 = tau*phidot+phi
```

```
c     CHECK FOR THRUSTER START/STOP

c     CHECK X-AXIS
      if ((tr1 .ge. db) .and. (flagxp .eq. (.false.))) then
        Mcx = -2.5*thrust
        Mcz = 2.5*thrust*dtan(zeta)
        flagxp = .true.
        flagxm = .false.
        call initham(nxt)
       endif
      if ((tr1 .lt. db) .and. (flagxp .eq. (.true.))) then
        mcx = 0
        Mcz = 0
        flagxp = .false.
        call initham(nxt)
       endif
      if ((tr1 .le. (0-db)) .and. (flagxm .eq. (.false.))) then
        Mcx = 2.5*thrust
        Mcz = -2.5*thrust*dtan(zeta)
        flagxm = .true.
        flagxp = .false.
        call initham(nxt)
       endif
      if ((tr1 .gt. (0-db)) .and. (flagxm .eq. (.true.))) then
        Mcx = 0
        Mcz = 0
        flagxm = .false.
        call initham(nxt)
       endif

c     CHECK FOR PERTURBATION DISCONTINUITIES
      b1= 1.047197551
      b2 = 0.017453292
      if (dmod((x(10,nxt)+b2),b1) .le. (2*b2)) then

c     Manually Propagate State Vector
      Phin = phi+h*phidot
      phidotn = phidot+h*F(4,nxt)
      thetan = theta+h*thetadot
      thetadotn = thetadot + h*F(5,nxt)
      psin = psi+h*psidot
      psidotn = psidot+h*F(6,nxt)
      nun = x(10,nxt)+h*f(10,nxt)
      alphan= nun+alpha0
```

```
c     Determine next Solar unit vector
      Sxn = dcos(thetan)*dcos(psin)*dsin(alphan)
     1     *dcos(delta)+dcos(thetan)
     2     *dsin(psin)*dsin(delta)-dsin(thetan)*dcos(alphan)*dcos(delta)
      Syn = (dsin(thetan)*dsin(phin)*dcos(psin)-dcos(phin)*dsin(psin))
     1     *dsin(alphan)*dcos(delta)+(dcos(phin)*dcos(psin)+dsin(phin)
     2     *dsin(thetan)*dsin(psin))*dsin(delta)+dsin(phin)*dcos(thetan)
     3     *dcos(alphan)*dcos(delta)
      Szn = (dsin(phin)*dsin(psin)+dcos(phin)*dsin(thetan)*dcos(psin))
     1     *dsin(alphan)*dcos(delta)+(dcos(phin)*dsin(thetan)*dsin(psin)
     2     -dsin(phin)*dcos(psin))*dsin(delta)+dcos(phin)*dcos(thetan)
     3     *dcos(alphan)*dcos(delta)

c     Check to see if crossing occurs
      z = 0
      if ((sxn*sx) .le. (0.0)) then
        z=z+1
        endif
      if ((Syn*Sy) .le. 0.0) then
        z = z+1
       endif
      if (((Sxn/2+0.8660254*Szn)*(Sx/2+0.8660254*Sz)) .le. 0.0) then
        z=z+1
       endif
      if (((Sxn/2-0.8660254*Szn)*(Sx/2-0.8660254*Sz)) .le. 0.0) then
        z=z+1
       endif

c     If crossing occurs, Reinit Haming with Propagated state
      if (z .gt. 0) then
       x(1,1) = phin
       x(4,1) = phidotn
       x(2,1) = thetan
       x(5,1) = thetadotn
       x(3,1) = psin
       x(6,1) = psidotn
       x(10,1) = nun
       t = t+h
       nxt = 0
       call haming(nxt)
       if (nxt .ne. 0) go to 91
        write(15,*) 'Haming bombed'
        stop
```

```
 91  continue
     endif

     endif

c    STEP HAMING THROUGH TO NEXT TIME INCREMENT
     call haming(nxt)

c    CHECK LIMITS AND TOTAL THRUSTER ACTIVITY
      if (dabs(phi) .gt. (db)) then
        outx = outx+h
       else
        inx = inx+h
       endif
      if (dabs(theta) .gt. (db)) then
        outy = outy+h
       else
        iny = iny+h
       endif
      if (dabs(psi) .gt. (db)) then
        outz = outz+h
       else
        inz = inz+h
       endif
      if (Mcx .eq. 0.0) then
        offX = offX+h
       else
        onx = onx+h
       endif

      if ((dabs(theta) .gt. (db)) .or.
     1    ((dabs(psi) .gt. (db)) .or. (dabs(phi) .gt. (db)))) then
       outtot = outtot+h
       endif

c      GRAPHING DATA OUTPUT BLOCK
c       if (tt .ge. 500) then
c        write(15,*) ct,phi,phidot,theta,thetadot,psi,psidot,f(4,nxt)
c        write(14,*) wx,wy,wz,xblock,yblock,zblock
c        write(15,*) psi,theta,phi
c        tt = 0
c       endif
c     tt = tt+h
      ct = ct + h
```

```
      goto 200
      endif
  100 continue
      NumFire = onX+onY+onZ
      time = ct
      percentx = onx/time
      percenty = ony/time
      percentz = onz/time
      outx=outx/time
      outy=outy/time
      outz=outz/time
      outtot = outtot/time
      percentt = percentx+percenty+percentz
      timp = percentt*tf*2.5*thrust
      write(14,41) thrust,time,outx,outy,outz,outtot
      write(15,44) thrust,percentt,timp,hy
   41   format(f10.8,f12.2,4(x,f10.8))
   44   format(2(x,f10.8),x,f12.4,x,f10.6)
  300 continue
      close(14)
      close(15)
      stop
      end
c
c
      SUBROUTINE initham(nxt)
c
c     REINITIALIZE HAMING AFTER THRUSTER START/STOP
c
      implicit double precision(a-h,m,o-z)
      double precision Ix,Iy,Iz,nu,thrusts(12)

      common /ham/ t,x(12,4),f(12,4),errest(12),n,h
      common /wheel/ hx,hy,hz,hxdot,hydot,hzdot
      common /const/ Ix,Iy,Iz,w0,w0dot,ecc
      common /state/ phi,theta,psi,phidot,thetadot,psidot
      common /moments/ db,thrust,mcx,mcy,mcz,mx,my,mz
      common /sun/ alpha0,delta,Sx,Sy,Sz
      common /nominal/ phi0,theta0,psi0

      x(1,1) = phi
      x(2,1) = theta
      x(3,1) = psi
      x(4,1) = phidot
```

```
      x(5,1) = thetadot
      x(6,1) = psidot
      x(7,1) = hx
      x(8,1) = hy
      x(9,1) = hz
      x(10,1) = x(10,nxt)
      nxt = 0
      if ((mcx .eq. 0)) then
        h = 0.5
       else
        h = 0.1
       endif
      call haming(nxt)
      if(nxt .ne. 0) go to 51
       write (15,*) ' haming did not reinitialize'
      NumFire = onX+onY+onZ
      time = ct

      stop
   51 continue
      return
      end

      SUBROUTINE perts(nxt)
c
c     ****************************************
c        Determines perturbative effects
c     ****************************************
c
      implicit double precision (a-h,m,o-z)
      double precision Ix,Iy,Iz,nu
      common /ham/ t,x(12,4),f(12,4),errest(12),n,h
      common /const/ Ix,Iy,Iz,w0,w0dot,ecc
      common /state/ phi,theta,psi,phidot,thetadot,psidot
      common /moments/ db,thrust,Mcx,Mcy,Mcz,Mx,My,Mz
      common /sun/ alpha0,delta,Sx,Sy,Sz

C     DETERMINE SATELLITE MEAN MOTION AND ANGULAR VELOCITY

      meanmot = 2*3.1415962/((23*60+56)*60)
      nu = x(10,nxt)
      w0 = meanmot*(1+ecc*dcos(nu))**2/((1-ecc*ecc)**1.5)
      f(10,nxt) = w0
      mu = meanmot**2.0*3*((1+ecc*dcos(nu))/(1-ecc*ecc))**3.0
```

```
c
c     GRAVITATIONAL TORQUES
c
      Mgx = mu*(Iz-Iy)*dsin(phi)*dcos(phi)*dcos(theta)
     1    *dcos(theta)
      Mgy = mu*(Iz-Ix)*dsin(theta)*dcos(theta)
     1    *dcos(phi)
      Mgz = mu*(Ix-Iy)*dsin(theta)*dcos(theta)*dsin(phi)
c
c     SOLAR PRESSURE TORQUES
c
      A1 = 1.25*1.25*3.1415962
      A2 = 0.75*0.75*3.1415962
      A3 = 3.0
      A4 = 3.0
      Ps = 4.644D-6
      rho = 0.9
      rt3 = sqrt(3.0)
      alpha = alpha0+nu

c     DETERMINE SOLAR UNIT VECTOR IN BODY FRAME COORDINATES

      Sx = dcos(theta)*dcos(psi)*dsin(alpha)*dcos(delta)+dcos(theta)
     1     *dsin(psi)*dsin(delta)-dsin(theta)*dcos(alpha)*dcos(delta)
      Sy = (dsin(theta)*dsin(phi)*dcos(psi)-dcos(phi)*dsin(psi))
     1     *dsin(alpha)*dcos(delta)+(dcos(phi)*dcos(psi)+dsin(phi)
     2     *dsin(theta)*dsin(psi))*dsin(delta)+dsin(phi)*dcos(theta)
     3     *dcos(alpha)*dcos(delta)
      Sz = (dsin(phi)*dsin(psi)+dcos(phi)*dsin(theta)*dcos(psi))
     1     *dsin(alpha)*dcos(delta)+(dcos(phi)*dsin(theta)*dsin(psi)
     2     -dsin(phi)*dcos(psi))*dsin(delta)+dcos(phi)*dcos(theta)
     3     *dcos(alpha)*dcos(delta)

c     CALCULATE SOLAR PRESSURE TORQUES

      Msx = -Ps*(2*A3*dabs(Sy)*(1+rho)*Sy+2*A4*dabs(Sx)*(1-rho)*Sy)
      Msy = Ps*(-3.4*A1*dabs(0-Sx/2+rt3/2*Sz)*(Sz*(1+rho/2)
     1    -rt3/2*rho*Sx)+2.85*A2*dabs(Sx/2+rt3/2*Sz)
     2    * ((1+rho/2)*Sz-rt3/2*rho*Sx)+2*A3*dabs(Sy)*(1-rho)*Sx
     3    +2*A4*dabs(Sx)*(1+rho)*Sx)
      Msz = Ps*(3.4*A1*dabs(-Sx/2+rt3/2*Sz)*(1-rho)*Sy
     1    -2.85*A2*dabs(Sx/2+rt3/2*Sz)*(1-rho)*Sy)
c
c     CALCULATE W0DOT
```

```
c
      w0dot = 0-2*meanmot*meanmot*ecc*dsin(nu)/(1+ecc*dcos(nu))**3
      w0dot = w0dot/((1-ecc*ecc)**3)
c
c     CALCULATE RADIO-FREQUENCY TORQUES
c
      Mtx = 0 0
      Mty = 5.667d-6
      Mtz = 0.0
c
c     SUM DISTURBANCE TORQUES
c
      Mx = Mgx+Msx+Mtx
      My = Mgy+Msy+Mty
      Mz = Mgz+Msz+Mtz
      return
      end
c
      SUBROUTINE rhs(nxt)
c
c     ************************************************************
c     rhs is the right hand side of the differential equations.
c     ************************************************************
c
      implicit double precision (a-h,m,o-z)
      double precision Ix,Iy,Iz
      common /ham/ t,x(12,4),f(12,4),errest(12),n,h
      common /const/ Ix,Iy,Iz,w0,w0dot,ecc
      common /wheel/ hx,hy,hz,hxdot,hydot,hzdot
      common /state/ phi,theta,psi,phidot,thetadot,psidot
      common /moments/ db,thrust,Mcx,Mcy,Mcz,Mx,My,Mz
      common /dbug/ xblock,yblock,zblock,wx,wy,wz

c
c     w0 = satellite mean motion
c     w0dot = rate of change of w0
c     phi = roll          :x(1)
c     theta = pitch       :x(2)
c     psi = yaw           :x(3)
c     phidot, thetadot, and psidot are first derivatives
c     Mx, My, Mz are moments about each axis
c     hx,hy,hz are reaction wheel moments, and __dot are their rates of
c     change
c
```

```
c     Set new values of Phi,Theta, Psi and dot terms
c
      phi =   x(1,nxt)
      theta = x(2,nxt)
      psi =   x(3,nxt)
      phidot =   x(4,nxt)
      thetadot = x(5,nxt)
      psidot =   x(6,nxt)
      hx = x(7,nxt)
      hy = x(8,nxt)
      hz = x(9,nxt)
c
c     CALL SUBROUTINES TO DETERMINE MOMENTS
c
      call perts(nxt)
      Mx = Mx+Mcx
      My = My+Mcy
      Mz = Mz+Mcz
      hydot=592*(5*thetadot+theta)

c
c     wx,wy,wz are the rotation of the body frame wrt inertial space
c
      wx = phidot-psidot*dsin(theta)-w0*dcos(theta)*dsin(psi)
      wy = thetadot*dcos(phi)+psidot*dcos(theta)*dsin(phi)-w0*(dcos(phi)
     1     *dcos(psi)+dsin(phi)*dsin(theta)*dsin(psi))
      wz = psidot*dcos(theta)*dcos(phi)-thetadot*dsin(phi)+w0*(dsin(phi)
     1     *dcos(psi)-dcos(phi)*dsin(theta)*dsin(psi))
c
c     F(1) = PHI'
   1  F(1,nxt)=PHIDOT
C     F(2) = THETA'
   2  F(2,nxt)=THETADOT
C     F(3) = PSI'
   3  F(3,nxt) = PSIDOT
C     F(4) = PHI"
C     F(5) = THETA"
C     F(6) = PSI"
      xblock = (Mx-wy*wz*(Iz-Iy)+wz*hy)/Ix
      yblock = (My-hydot-wx*wz*(Ix-Iz))/Iy
      zblock = (Mz-wy*wx*(Iy-Ix)-wx*hy)/Iz
   4  F(4,nxt)=xblock+dsin(phi)*dtan(theta)*yblock
     1     +dcos(phi)*dtan(theta)*zblock
     2     +w0dot*dsin(psi)/dcos(theta)
```

```
     3      +w0*(psidot*dcos(psi)/dcos(theta)
     4      +phidot*dtan(theta)*dcos(psi))
     5      +thetadot*psidot/dcos(theta)
     6      +thetadot*phidot*dtan(theta)
  5  F(5,nxt)=dcos(phi)*yblock-dsin(phi)*zblock
     1      +w0dot*dcos(phi)
     2      +w0*(phidot*dsin(theta)*dsin(psi)-psidot*dsin(psi))
     3      -phidot*psidot*dcos(theta)
  6  F(6,nxt)=(dsin(phi)*yblock+dcos(phi)*zblock
     1      +w0*(thetadot*dcos(theta)*dsin(psi)-phidot*dcos(psi)
     2      +psidot*dsin(theta)*dcos(psi))
     3      +w0dot*dsin(theta)*dsin(psi)
     4      +phidot*thetadot+thetadot*psidot*dsin(theta))
     f(6,nxt)=F(6,nxt)/(dcos(theta))
  7  F(7,nxt)=0
  8  F(8,nxt)=hydot
  9  F(9,nxt)=0
     return
     end
c
     SUBROUTINE HAMING(NXT)

* Version of 11/07/90
* Purpose
*    Subroutine for integrating a system of first order differential
*    equations.  It is a fourth order predictor-corrector algorithm
*    which means it carries the last four values of the state vector,
*    and extrapolates these values to obtain a predicted next value
*    (the prediction step) and evaluates the equations of motion at
*    the predicted point, and then corrects the extrapolated point
*    using a higher order polynomial (the correction step).
* Input
*    NXT = specifies which of the four values of the state vector is
*          the current one.  NXT is updated by HAMING automatically,
*          but must be set to ZERO on the first call.
* Call Subroutines
*    RHS(NXT) = evaluates the equations of motion
* External Functions
*    None
* Common Blocks
*    HAM = Memory block shared by the main driver and subroutine RHS.
*    The common block contains:
*      X       = is the independent variable (often time)
*      Y(MAX,4) = the state vector (4 copies), with NXT pointing to
```

```
*               the current one, the limit of MAX EOM can be changed
*               through the PARAMETER in main driver, sub program
*               RHS, and below.
*       F(MAX,4) = are the EOM evaluated at the same times as the state
*               vector Y ... it is the job of sub program RHS to
*               calculate these.
*       ERR(MAX) = is an estimate of the one-step integration error
*       N       = is the number of ODES ... limit is MAX unless you change
*               the PARAMETER statement in main driver, sub program
*               RHS, and below.
*       H       = is the timestep ... one call to HAMING increments X by H
* References
*     Donald G. M. Anderson -- Harvard (1972)
* Analysis
*     William Wiesel - AFIT
* Programmer
*     Rodney D. Bain - AFIT
* Program Modifications
*     Original program slightly modified by Weisel and Bain.
* Comments
*     TOL = is HAMING's startup tolerance ... set to reasonable value
*         as necessary in PARAMETER statement.
*     The user must supply a main driver, and the subroutine RHS(NXT)
*     which evaluates the equations of motion.

      IMPLICIT REAL*8 (A-H,O-Z)    ! Global double precision
      PARAMETER (ZERO=0.D0, ONE=1.D0, TWO=2.D0, THREE=3.D0,
     1          FOUR=4.D0, MAX=12, TOL=1.D-12)
      COMMON /HAM/ X,Y(MAX,4),F(MAX,4),ERR(MAX),N,H

* Check if this is the first call ... HAMING (like all predictor-
* correctors) needs 'previous' values

      IF(NXT) 190,10,200

* It is a forward Picard iteration (slow and expensive) to step
* backwards in time three steps to get the 4 previous points.  A
* successful startup returns NXT=1, and time has not been
* incremented.  If startup fails, NXT will be returned as ZERO.

 10   XO=X
      HH=H/TWO
      CALL RHS(1)
      DO 40 L=2,4
```

```
      X=X+HH
        DO 20 I=1,N
20          Y(I,L)=Y(I,L-1)+HH*F(I,L-1)
      CALL RHS(L)
      X=X+HH
        DO 30 I=1,N
30          Y(I,L)=Y(I,L-1)+H*F(I,L)
40     CALL RHS(L)
    JSW=-10
50  ISW=1
    DO 120 I=1,N
      HH=Y(I,1)+H*(9.D0*F(I,1)+19.D0*F(I,2)-5.D0*F(I,3)
   1     +F(I,4))/24.D0
    IF( DABS(HH-Y(I,2)) .LT. TOL) GOTO 70
    ISW=0
70  Y(I,2)=HH
    HH=Y(I,1)+H*(F(I,1)+FOUR*F(I,2)+F(I,3))/THREE
    IF( DABS(HH-Y(I,3)) .LT. TOL) GOTO 90
    ISW=0
90  Y(I,3)=HH
    HH=Y(I,1)+H*(THREE*F(I,1)+9.D0*F(I,2)+9.D0*F(I,3)
   1   +THREE*F(I,4))/8.D0
    IF( DABS(HH-Y(I,4)) .LT. TOL) GOTO 110
    ISW=0
110 Y(I,4)=HH
120 CONTINUE
    X=XO
    DO 130 L=2,4
      X=X+H
130    CALL RHS(L)
    IF(ISW) 140,140,150
140 JSW=JSW+1
    IF(JSW) 50,280,280
150 X=XO
    ISW=1
    JSW=1
    DO 160 I=1,N
160    ERR(I)=ZERO
    NXT=1
    GOTO 280

* A call to HAMING with NXT=-NXT, after a successful startup,
* will turn off the second evaluation of the equations of motion
* following the corrector step.  In systems where the equations of
```

```
* motion are very expensive, this can halve your run time.

190  JSW=2
     NXT=IABS(NXT)

* This is the predictor-corrector algorithm ... first the indices
* are premuted.

200  X=X+H
     NP1=MOD(NXT,4)+1
     GOTO (210,230),ISW
210  GOTO (270,270,270,220),NXT
220  ISW=2
230  NM2=MOD(NP1,4)+1
     NM1=MOD(NM2,4)+1
     NPO=MOD(NM1,4)+1

* ... then the predictor part is run to find an extrapolated value
* of the state vector at the new time ...

     DO 240 I=1,N
       F(I,NM2)=Y(I,NP1)+FOUR*H*(TWO*F(I,NPO)-F(I,NM1)
    1          +TWO*F(I,NM2))/THREE
240    Y(I,NP1)=F(I,NM2)-0.925619835D0*ERR(I)

* The equations of motion are evaluated at the extrapolated value
* of the state vector ...

     CALL RHS(NP1)

* and the corrector algorithm is used to add this new information
* and obtain a better value of the new state vector ...

     DO 250 I=1,N
       Y(I,NP1)=(9.D0*Y(I,NPO)-Y(I,NM2)+THREE*H*(F(I,NP1)
    1          +TWO*F(I,NPO)-F(I,NM1)))/8.D0
       ERR(I)=F(I,NM2)-Y(I,NP1)
250    Y(I,NP1)=Y(I,NP1)+0.0743801653D0*ERR(I)
     GOTO (260,270),ISW

* Finally, the equations of motion are re-evaluated at the better
* value of the state vector ... this can be suppressed.

260  CALL RHS(NP1)
```

```
270  NXT=NP1

280  RETURN
     END
```

Bibliography

1. Agrawal, Brij N. *Design of Geosynchronous Spacecraft.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1986.

2. Aston, G. and J.R. Brophy. "A Detailed Model of Electrothermal Propulsion Systems," *AIAA/ASME/SAE/ASEE 25th Joint Propulsion Conference.* No. 89-2262. New York: American Institute of Aeronautics and Astronautics, July 1989.

3. Bate, Roger B. and others. *Fundamentals of Astrodynamics.* New York: Dover Publications, Inc., 1971.

4. Beattie, J.R. and others. "Xenon Ion Propulsion Subsystem," *Journal of Spacecraft Power and Propulsion, 5*: 438-444 (July-Aug 1989).

5. Brophy, J.R. and G. Aston. "A Detailed Model of Ion Propulsion Systems," *AIAA/ASME/SAE/ASEE 25th Joint Propulsion Conference.* Paper No. 89-2268. New York: American Institute of Aeronautics and Astronautics, July 1989.

6. Burton, R.C. and others. "Theory of the Pulsed Electrothermal Thruster," *AIAA/JSASS/DGLR 16th International Electric Propulsion Conference.* Paper No. 82-1952. New York: American Institute of Aeronautics and Astronautics, November 1982.

7. Burton, R.C. and others. "Experiments on a Repetitively Pulsed Electrothermal Thruster," *Journal of Spacecraft Power and Propulsion, 6*: 139-144 (March-April 1980).

8. Day, M.L. and others. "Intelsat VII Ion Propulsion Subsystem Implementation Study," *AIAA/DGLR/JSASS 21st International Electric Propulsion Conference.* Paper No. 90-2550. New York: American Institute of Aeronautics and Astronautics, July 1990.

9. Dougherty, H.J. and others. "Attitude Stabilization of Synchronous Communications Satellites Employing Narrow-Beam Antennas," *Journal of Spacecraft and Rockets, 8*: 834-841 (August 1971).

10. Ghislanzoni, L. and others. "The Application of Arcjet Propulsion Systems for Geostationary Satellites," *Proceedings of the 21st International Electric Propulsion Conference.* Paper No. 90-2548. New York: American Institute of Aeronautics and Astronautics, July 1990.

11. Groh, K.H. and H.W. Loeb. "State-of-the-Art of Radio-Frequency Ion Thrusters," *AIAA/ASME/SAE/ASEE 25th Joint Propulsion Conference.* Paper 89-2381. New York: American Institute of Aeronautics and Astronautics, July 1989.

12. Hirata, Mashiro and Hiroshi Murakami. "Impulse Measurement of Pulsed Plasma Engine," *AIAA/JSASS/DGLR 16th International Electric Propulsion Conference.* Paper 82-1875. New York: American Institute of Aeronautics and Astronautics, November 1982.

13. Nabi, T.A. and others. "Intelsat VII Communications Capabilities and Performance," *Proceedings of the 13th International Communications Satellite Systems Conference.* Paper No. 90-0787. New York: American Institute of Aeronautics and Astronautics, 1990.

14. Nakamura, Y. and others. "Performance Design of a 10-mN Ion Thruster," *AIAA/JSASS/DGLR 16th International Electric Propulsion Conference.* Paper 82-1916. New York: American Institute of Aeronautics and Astronautics, November 1982.

15. Neyret, P. and others. "The Intelsat VII Spacecraft," *Proceedings of the 13th International Communications Satellite Systems Conference.* 95-110. Paper No. 90-0788. New York: American Institute of Aeronautics and Astronautics, 1990.

16. Planeaux, Capt James. Class lecture in MENG 632, Nonchemical Rocket Propulsion. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1991.

17. Sovey, James S. and David J. Pidgeon. "Advanced Propulsion for Leo and GEO Platforms," *AIAA/DGLR/JSASS 21st International Electric Propulsion Conference.* Paper No. 90-2551. New York: American Institute of Aeronautics and Astronautics, July 1990.

18. Suzuki, Hiroshi and Kyoichi Kuriki. "Fast Acting Valve for a Quasi-Steady MPD Arcjet," *AIAA/JSASS/DGLR 16th International Electric Propulsion Conference.* Paper 82-1886. New York: American Institute of Aeronautics and Astronautics, November 1982.

19. Valentian, D. and others. "Application of Field Emission Thruster to Spacecraft Fine Attitude Control and Orbit Positioning," *AIAA/ASME/SAE/ASEE 25th Joint Propulsion Conference.* Paper No. 89-2277. New York: American Institute of Aeronautics and Astronautics, July 1989.

20. Wilson, Andrew, ed. *Interavia Space Directory, 1990-1991*. Alexandria, Va: Janes Information Group, 1991.

Vita

Jeffrey J. Abbott was born in Holyoke, Massachussetts on 29 September 1962. He was the sixth of seven children of Darleen and John Abbott. He graduated from Robert E. Lee High School in 1980 and attended the United States Air Force Academy where he graduated in 1984 with a Bachelor of Science in Astronautical Engineering. He was then commissioned as a 2nd Lieutenant in the USAF and assigned to Sunnyvale AFS, California, as a trainee in satellite operations. He transferred with his organization to Peterson AFB, Colorado, where the unit was activated as the 1st Satellite Control Squadron. He performed duties as a Satellite Vehicle Engineer at Peterson AFB and Falcon AFB until May of 1990, when he entered the Air Force Institute of Technology. After graduation, he will be assigned to Space Systems Division, Los Angeles AFB, California.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 91	Masters's Thesis

4. TITLE AND SUBTITLE
APPLICATION OF LOW THRUST PROPULSION TECHNIQUES TO SATELLITE ATTITUDE CONTROL SYSTEMS

5. FUNDING NUMBERS

6. AUTHOR(S)
Jeffrey J. Abbott, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GA/ENY/91D-9

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Equations of motion for a satellite controlled through continuous, low thrust propulsion systems are analyzed through numerical integration techniques. The equations of motion are derived using the Euler Moment Equations. The properties of the satellite model are based upon the Intelsat VII design of communications satellite. A simple rate and error feedback controller is used in providing active attitude control about two and three axis. Perturbation models are created and applied based upon the satellite model. Parameters varied include thrust levels and controller deadband widths. System response times, pointing accuracy, and total impulse required for attitude control are determined as measurements of relative performance.

14. SUBJECT TERMS
Satellite Attitude Control, Electric Propulsion, Low Thrust Propulsion

15. NUMBER OF PAGES
107

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL